commodore microcomputers

July/August 1984
$2.50 U.S.
$3.50 Canada
ISSN 0744-8724

Opening the Door to
TELECOMMUNICATIONS

USE YOUR COMPUTER
TO SHOP,
BANK,
VACATION,
"MEET" OTHERS,
LEARN COMPUTING
AND MUCH MORE

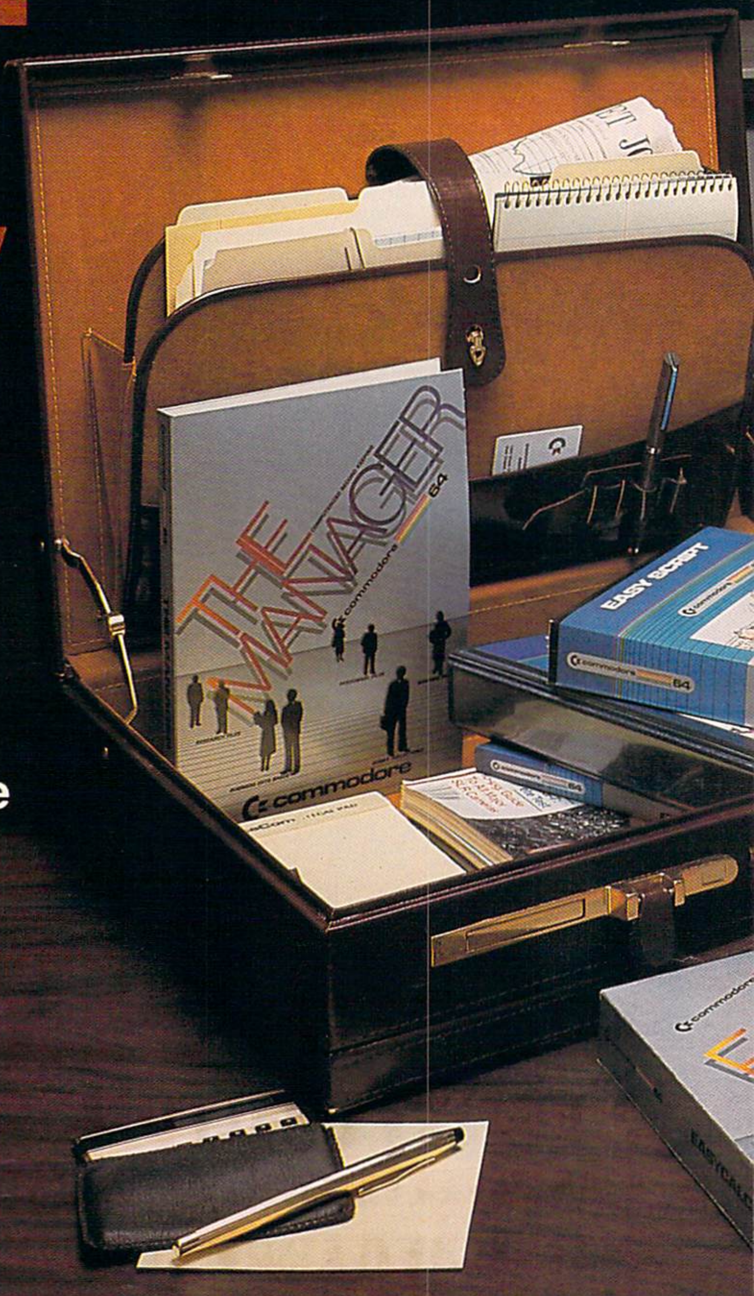
BULK RATE
U.S. POSTAGE
PAID
Permit No. 1094
St. Cloud, MN 56301

Commodore Business Machines, Inc.
1200 Wilson Drive • West Chester, PA 19380

Address Correction Requested

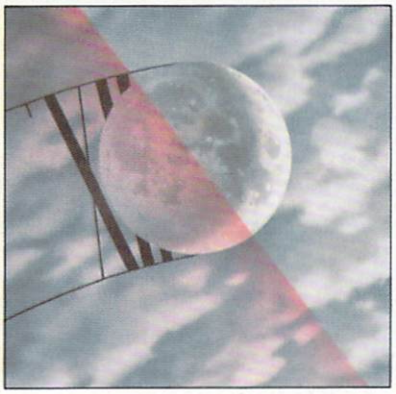
Commodore Indispensable Software... For When You Really Mean Business

Computer Systems don't have to be expensive to be effective —even in business. Commodore has been proving that for years.



commodore microcomputers

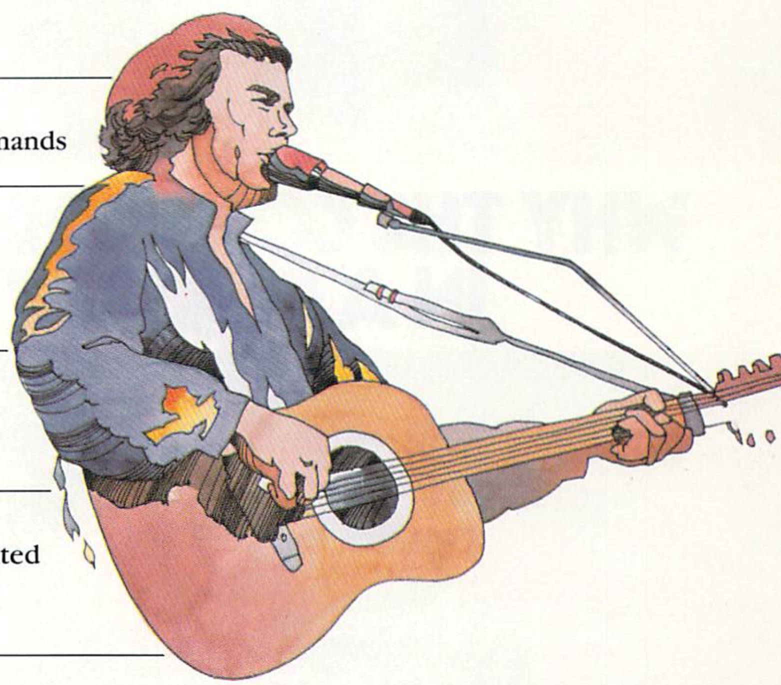
Volume 5, Number 3, Issue 30



features

using your computer

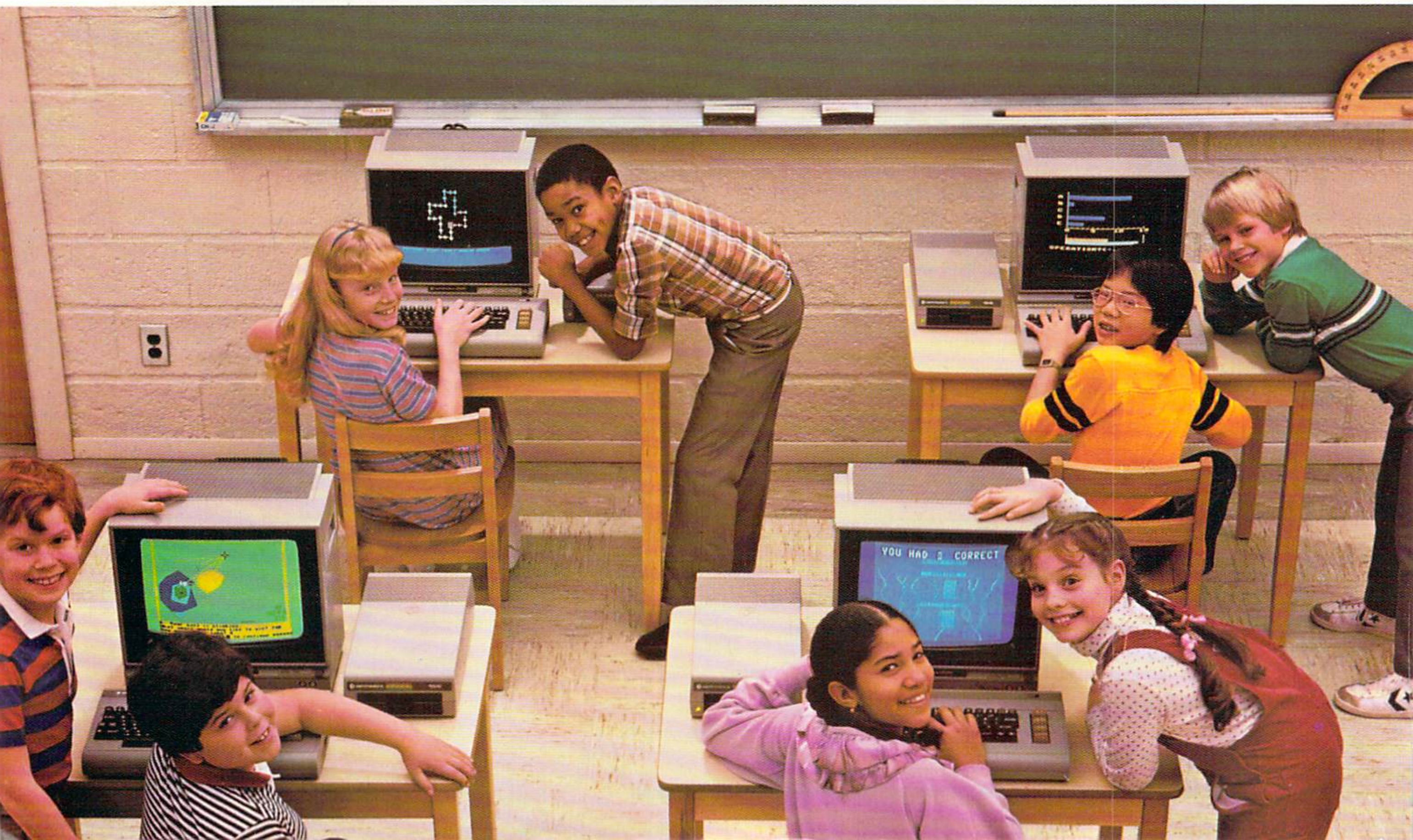
understanding your computer



reviews

departments

WHY THE COMMODORE 64 ISN'T IN A CLASS BY ITSELF.



Director—Publishing
Neil Harris

Editor
Diane LeBold

Technical Editor
Jim Gracely

Assistant Editor
Carol Minton

Staff Writers
Tony Caramanico
Larry Greenley
Barbara Karpinski
Thomas Ziegler

Contributing Writers
Francis V. Amato, Dick Barnes, Mark Brownstein, M. W. Caprio, deb! Christensen, Philip S. Dale, Hugh Doss, Kelley M. Essoe, Robert Felice, Al Fragola, Shlomo Ginsburg, Richard H. Goodyear, Donald E. Hassler, Craig R. Hessel, Rick Jeandell, Joe Rotello, Herbert Swartz, Jack Weaver, Richard Winters, Mark Zimmermann

Technical Staff
Tony Caramanico
Barbara Karpinski

Circulation Manager
John O'Brien

Circulation Assistant
Kathy Reigel

Advertising Sales Manager
Pamela S. Fedor

Advertising Coordinator
Sharon Steinhofer

Graphic Design
Neumann, Greenberg, Schlenker
King of Prussia, Pennsylvania

Cover
Verlin Miller

Typography
Associates International, Inc.
Wilmington, Delaware

Printing
Volkmuth Printers
St. Cloud, Minnesota

ISBN 0-88731-010-9

Watch for These Upcoming Issues

Power/Play, Issue 10 (August/September): Next issue we're featuring user groups—who's joining them and why, how to get one started, where to find them and what they do. If you've ever wished you had someone to ask about that program you just can't get to work, needed some advice about which printer to buy or just plain wanted some friends you could talk to about computing, DON'T MISS THIS ISSUE!

Commodore Microcomputers, Issue 31 (September/October): Productivity software is our feature topic next issue. Find out about those word processors, data bases, spread sheets and accounting packages for your computer. Which ones are best for your needs? We'll help you sort it out in September.

Key to Entering Program Listings

```
"[F1,F2,F3,F4,F5,F6,F7,F8]":F1,F2,F3,F4,
  F5,F6, F7 AND F8
"[POUND]":ENGLISH POUND
"[PI]"PI SYMBOL
"^":UP ARROW
"[HOME]":UNSHIFTED CLR/HOME
"[CLEAR]":SHIFTED CLR/HOME
"[RVS]":REVERSE ON
"[RVOFF]":REVERSE OFF
"[BLACK,WHITE,RED,CYAN,MAGENTA,GREEN,BLUE,
  YELLOW]" THE 8 CTRL KEY COLORS
"[ORANGE,BROWN,L. RED,GRAY 1,GRAY 2,L.
  GREEN,L. BLUE, GRAY 3]":THE 8
  COMMODORE KEY COLORS (ONLY ON THE 64)
GRAPHIC SYMBOLS WILL BE REPRESENTED AS
  EITHER THE LETTERS SHFT (SHIFT KEY) AND
  A KEY: "[SHFT Q,SHFT K,SHFT V,SHFT T,
  SHFT L]"
  OR THE LETTERS CMDR (COMMODORE KEY) AND
  A KEY:"[CMDR Q,CMDR H,CMDR S,CMDR N,
  CMDR O]"
IF A SYMBOL IS REPEATED, THE NUMBER OF
  REPITITIONS WILL BE DIRECTLY AFTER THE
  KEY AND BEFORE THE COMMA:"[SPACE3,
  SHFT S4,CMDR M2]"
```


Commodore: The Microcomputer Magazine is published by the Computer Systems Division, Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, Pennsylvania 19380.

Subscription Orders: Phone 800-345-8112 (In Pennsylvania 800-662-2444).

Go Even More Directly to XY

To the Editor:

Your publication is always informative, but I was particularly drawn to the article "Go Directly to XY!" on page 111 of Volume 4, Number 6, as I had myself already struggled with, and solved, the problem of the elusive PRINT AT on the Commodore 64, and can therefore suggest, within the framework of the testbed program listed in that article (and employing its somewhat unorthodox convention of letting X represent the row, and Y the column), the following subroutine:

```
10 IF X THEN POKE 214,
   X−1:PRINT:PRINT TAB (Y);:
   RETURN
20 PRINT CHR$(19)TAB(Y);:
   RETURN
```

This uses 71 bytes over and above the testbed, and takes an average of 39.89 seconds to run. However two points should be noted:

1) On my machine the time to run using the "horrible example" subroutine on page 111 averages 216.92 seconds, not the 252.64 seconds cited. This suggests that the 64 is inherently faster than the PET/CBM, and that a scaling factor of 252.64/216.92, or 1.16, should be applied to my figure of 39.89 seconds, resulting in 46.46 seconds, still a respectable figure, especially considering that no machine language is used.

2) Lines 130 and 140 of the testbed program on page 114 generate values of X from one to 12, and of Y from one to 80, which may for all I know be appropriate for a PET/CBM, but for a 64 I suggest the following:

```
130 X=INT(25*RND(0))
140 Y=INT(39*RND(0))
```

(39 rather than 40 to prevent scrolling)

With these lines substituted in the testbed program, execution time drops to an average of 35.58 seconds, unscaled.

John Auer
Willow Street, Pennsylvania

To the Editor:

Here is a way for VIC 20 users to "Go Directly to XY!" The technique relies on a built-in Kernal function named PLOT, which can both read and set the screen cursor position. For our purposes, calling this routine with the accumulator carry flag cleared moves the cursor to the screen row and column specified by the .X and .Y registers, respectively. The next PRINT statement executed will begin printing at the new cursor position.

To use the PLOT function, a short machine language program is required to clear the carry flag, load the .X and .Y registers, and then jump to the routine. The machine language and the statements which load it into the cassette buffer (out of the way of BASIC) are in the listing below.

All that is needed now is a simple BASIC subroutine that will POKE the X and Y coordinates (0-22 rows, 0-21 columns) into the machine language program before it is executed:

```
100 POKE 830,X: POKE 832,Y:
    SYS 828: RETURN
```

That's it! The method is memory-efficient and extremely fast—the BASIC subroutine takes only 24 bytes and the testbed used in the article ran in only 35.98 seconds.

Ron Tunin
Fort Wayne, Indiana

To the Editor:

Commodore 64 users who read David Bull's article "Go Directly to XY" may be interested to know of an even simpler PRINT AT technique for the '64.

RAM locations 214 and 211 hold the X and Y cursor positions set by a ROM subroutine at 58732 decimal. This routine will quickly position the cursor for printing without any additional machine language coding, even if the screen memory has been relocated.

For example, to print "Hi,

```
      CLC                ;Clear the carry flag
      LDX #X             ;Load the .X register
      LDY #Y             ;Load the .Y register
      JMP PLOT           ;Jump to PLOT function

10    FOR I = 828 TO 835
20    READ A: POKE I,A: NEXT I
30    DATA 24,162,0,160,0,76,240,255
```

Mom!" at row 11, column 23, we could code in BASIC:

```
5 GOTO 100
10 POKE214,X:POKE211,Y:
   SYS58732:RETURN
100 X=11:Y=23:GOSUB 10:PRINT
   "HI, MOM!"
```

The subroutine in line 10 gives a time of 41.61 seconds and overhead of just 21 bytes when used in David Bull's testbed program.

Jan N. Pedersen, Jr.
Fresno, California

Charming Printing

To the Editor:

Came up with two variations on Andy Gamble's "Prints Charming" (Issue 27).

```
100 a$="back to front"
110 for i=len(a$) to 1
    step -1
120 ?tab(i)"  "mid$(a$,
    i,1);"CU"
130 next i

100 a$="        ripple"
105 for j=1 to 50
110 for i=len(a$)to 1
    step -1
120 print tab(i)"  "mid$
    (a$,i,1);"CU"
130 next i;next j
```

("CU" stands for cursor up)

Try these! You'll like them!

Thanks Andy!

Thomas Johnson
Villanova, Pennsylvania

Old ROM Variables

To the Editor:

This letter is in response to Jack B. Cooper's letter regarding the VAL($) function (Issue 26, page 11). Mr. Cooper wanted to know what values were assigned for certain variables on a model 2001 PET ("old" 2.0 ROMs). Here they are:

A$="E 1"	VAL(A$)=0
A$="E 38"	VAL(A$)=0
A$="E+99"	VAL(A$)=0
A$="E−99"	VAL(A$)=0
A$="E 100"	VAL(A$)=0
A$="E+9000"	VAL(A$)=OVERFLOW ERROR
A$="E−9000"	VAL(A$)=0
A$="E− 100"	VAL(A$)=0

Kim Moser
New York, New York

Yet Another Protection Scheme

To the Editor:

In response to David Williams' article "No List/Save" (Volume 4, Number 3, Issue 24) I find that a more elegant way to prevent a user from LISTing or SAVEing a program on the Commodore 64 can be achieved by POKEing 808,232. This statement changes the system vector which points to the routine handling RUN/STOP RESTORE warm starts. It will effectively prevent the user from stopping (and thereby listing and saving) a running program. If the program to be protected is of the type that will eventually end (instead of say, returning to the main menu), the statement POKE 775,168 will change the vector handling the LIST routine, causing an ?UNDEF'D STATEMENT ERROR if the user attempts to list all or part of the program. For more drastic measures, POKE 775,171 will cause the machine to crash on a list attempt. These methods may vary in effect depending on the program length, so it may be necessary to experiment with the contents of 775. It may not be necessary to use these last two techniques if you have used the first to disable the RUN/STOP RESTORE keys, as it will present a garbled listing of the program (it will still run OK). Lastly, you can prevent illegal saving of your program, without necessarily using the other techniques, by POKEing 819,246. Any attempt to SAVE will result merely in the READY message being displayed.

P.S. The default values for the aforementioned memory locations are: 775—167, 808—237, 819—245.

Adrian Boyle
Short Hills, New Jersey

The Opportunity for Change

Commodore's founding father, Jack Tramiel, summed it up pretty well when he said, "When you're through changing, you're through." I've always liked that philosophy, and it's certainly worked well for Commodore's computer division. You may have noticed with this issue that it seems to be working equally well for Commodore's publications.

For those of you who are new to our magazines (or who are simply unobservant) this issue does mark some significant changes for us—a new format, new title, new advertisers and some new staff, to mention the most obvious. The best part is that it's only the beginning (which is what we always say—because it's true). Over the next several issues you'll see bigger and better things from us as we continue to improve and expand.

The new format doesn't require much explanation. It's prettier, easier to follow and more exciting, thanks to the efforts of our excellent design team, who sweated out the details, working (as usual) against an impossible deadline. You might, however, wonder about the new title. Well, let me put it this way. It was getting to be a real pain in the neck referring to ourselves as *Commodore: The Microcomputer Magazine*, which we always shortened to *Commodore Magazine*, anyway. The new title, *Commodore Microcomputers*, combines the best of both titles, I think.

We're also happy to welcome several new advertisers, who are the first wave in our expanded advertising sales effort, now under the direction of our new advertising sales manager, Pamela Fedor. You can look forward to an increasing number of advertising pages over the next several issues—so you'll have more information about who's making what products for which Commodore computers, and where to get those products.

We've also added Carol Minton to our staff as assistant editor. Carol comes to us from, believe it or not, *TV Guide*, where she was cable editor. She's also a Commodore 64 enthusiast, who had been using her system at home before we ever heard of her. Carol will be mainly handling news and reviews—putting together our Industry News section and assigning products for in-depth review—which means *she'll* be the one buried up to her eyebrows in press releases, instead of me or Jim Gracely. This will be a great relief for us and Carol seems to like the idea as well.

So much for departmental scuttlebutt. Let's talk about our feature topic, telecommunications. Commodore has sold a lot of modems (the devices that allow you to communicate over telephone lines with your computer) and has been conducting an extremely successful data base called the Commodore Information Network via the CompuServe Information Service, a national telecommunications network. This has led us to believe that many of our users are actively involved in telecommunications. But what about the users who were left completely mystified by what I just said? For your sake, let me begin at the beginning.

If you get yourself a modem and the appropriate terminal software (that's terminal as in telephone or computer, not illness), you can hook up your computer to your touchtone telephone, dial the phone number of any one of a myriad of telecommunications services (like CompuServe, for instance) and begin receiving and/or transmitting information. I can't even begin to list the services that are available, since there are just too many, but I can tell you for starters that you can access Dow Jones News/Retrieval, shop at home, make airline reservations, read the World Book encyclopedia, do your schoolwork, get information on all kinds of products, learn computing (on the Commodore Information Network, for instance), play games with someone in another state, or just "chat" with a friend you may have never met in person. And that list barely scratches the surface of the things you can do.

Since you get a free subscription to CompuServe when you buy a Commodore modem, let's use that as a specific example. CompuServe is just one of several large national telecommunications net-

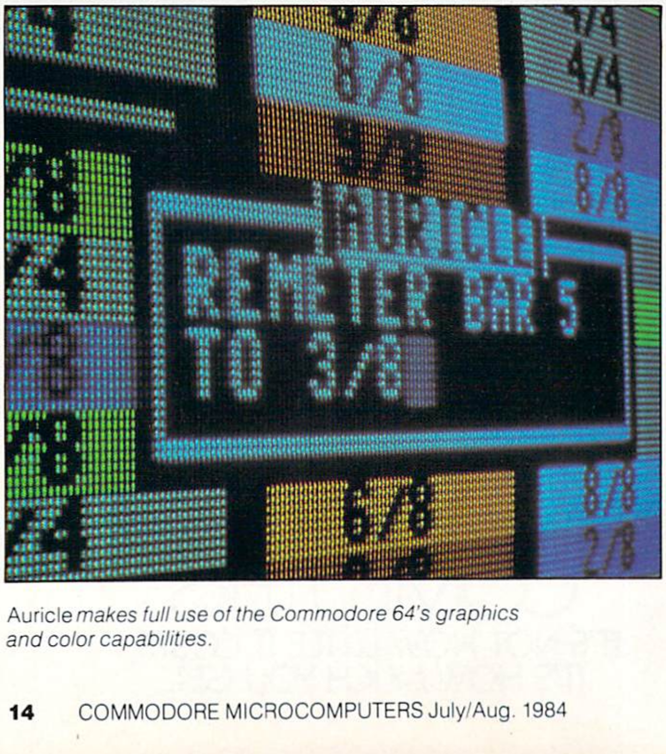
(Continued On Page 32)

Time Warp: Stardate 1984

by Kelley M. Essoe

Photo by Laurel Benton

Richard Grant (r.), creator of Auricle, *and his brother, television composer Ron Grant.*



Screen photos by Richard Benjamin Grant

Auricle *makes full use of the Commodore 64's graphics and color capabilities.*

Future Shock: Auricle *ushers in a whole new age of synergistic operating systems for computers... and computer users. So get ready to "beam up" to the future.*

Remember the computer system aboard the Starship Enterprise? Whenever the venerable Captain Kirk found himself in a tight spot and in urgent need of specifics on some glob-like invading life form or immediate strategic advice to help thwart the Klingons' endless assault on the rest of the God-fearing universe, he would consult the ship's computer.

"Computer, specifics on the glob-like invading life form please." And in the sexiest voice this side of Marilyn Monroe the computer would immediately respond with all the slimey details.

"How about a history of Klingon Commander Boy George?", Kirk asks, changing the subject. And without a blink of her mascara-ed memory chips or the audacity to refer him to a "main menu" in order to access another database, the computer would promptly disclose the complete and kinky life chronicle of Commander Boy George. Whatta gal, that computer.

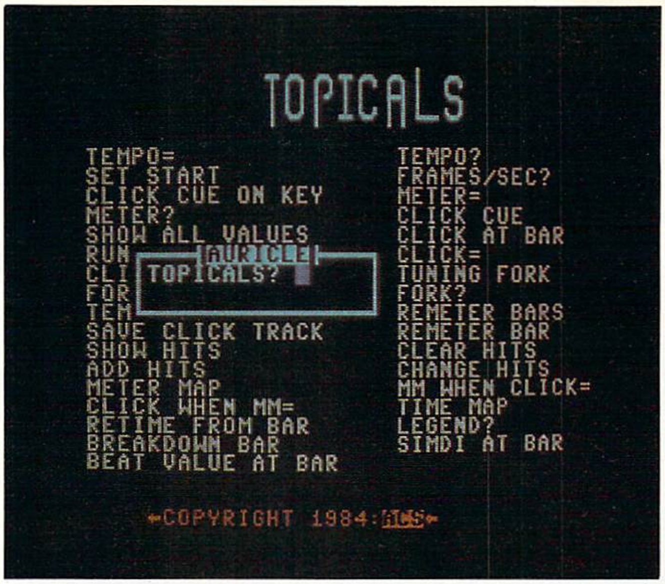
But then, of course, she is an extremely advanced cybernetic organism of the 25th century. Certainly by then our advancements in computer technology will, at the very least, allow us the kind of quick and simple repartee that Captain Kirk enjoyed with his computer.

He could communicate with her in English. Wonderfully handy. He could go from one order of business to another at the spur of the moment without blowing her dainty little circuits. That's nice. He could concurrently talk to her while she was talking to him, interjecting comments, requests and even changing the subject without having to either politely wait for her to finish or rudely shutting her up by slapping her sharply on her run/stop key. Now that's nifty. Best of all, he could access all her capabilities and knowledge instantly, without even so much as a fleeting glance at one single menu. All he had to do was ask and, on the spot, she delivered. (Which is more than the Post Office does right now and they have legs.)

That's what I call random access. What wouldn't we give to have a whack at her?

Unfortunately, the bad news is we don't have a chance in hades of living to meet her there in Stardate 43 point 6.

The good news is... we don't have to. Richard Grant, lawyer and computer hacker



"Topicals" are those words and phrases that make up the system's fundamental vocabulary. In the case of the Composer's Time Processor, *these words and phrases are, of course, related to music.*

par excellence, introduced me to her great-grandmother the other day.

Her name is *Auricle,* and she's every bit the liberated lady that her great-granddaughter on board the Enterprise will be. *Auricle* is, of course, quite young and has not yet matured to her full capacity. Nevertheless, her staggering potential is immediately obvious. It's like being introduced to a future queen. She's not wearing the jeweled crown yet, but she reeks of power and royalty. It's merely a matter of time.

At the moment she resides at the same address in which she was born: within a brand new, and in itself remarkable program for music composers called *Auricle: The Film Composer's Time Processor,* co-conceived by Richard Grant and his brother, composer Ron Grant, using a Commodore 64.

Auricle came into existence through the determination of the Grants to have their time processing program do exactly what they wanted it to do—as opposed to what the computer's operating system would or would not allow it to do.

Ron Grant, one of the alternating composers for the nighttime television drama "Knott's Landing", grew weary

Here the system is responding to a question concerning the "click" (tempo) at a specified bar.

"Anecdotals" are a subset of Topicals. It is here that the user can establish a personal "motif" for the functioning of the system.

of the painstaking and laborious task of figuring the crucial split-second timing involved in film scoring. A visionary of sorts, he knew there had to be a better way.

To a composer, movies or television are an endless stream of fractional seconds, each needing to be complemented by the mood-setting accompaniment of music. Throughout a film the music, like a shadow, must build and break in total unity with the action taking place on screen. Ideally, the musical tempo should speed up here, slow down there, imperceptibly snaking its way around in order to coordinate exactly with specific "hits", or dramatic points, within the storyline. But, until now, the composer had to find and settle for the most acceptable average within the rigid and uniform bounds of metronomic tempo. It's a risky business, for if not timed precisely, these hits quickly become misses, and the music becomes a premature clue or a too-late redundancy, either way standing out like a wart on the end of one's nose.

Sound synchronization is calculated by counting the frames and sprockets on a piece of film. Movie film is simply a long ribbon of single frames bordered on two sides by a chain of little square holes, or sprockets. The film itself looks like a series of separate photographs, which is exactly what it is.

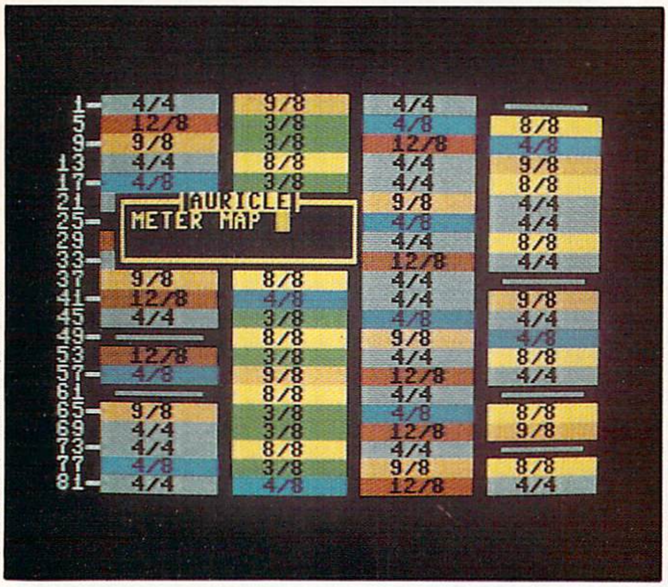
The speed at which the film races by in order to create the illusion of "real-time" motion is 24 frames per second. Accordingly, the traditional unit of time-frame measurement is 24 frames. Each frame has eight sprockets, providing the subunits to mark off fractions of frames. Therefore, a film composer is actually juggling three separate modes of time measurement: "real" time, frame/sprocket time, and metronomic time.

Take, for example, a hit occurring at two minutes and 42.7 seconds in real time. The composer must first convert the musical tempo that he wants to use into a frame count—usually some fraction of the standard 24 such as 19/4, or 19 frames plus 4 sprockets. From this working rate of cadence, he then has to adjust the music, fine tuning the tempo so that at precisely two minutes and 42.7 seconds the steady pulse of music climaxes in synchrony with the action taking place on screen.

So, like David valiantly facing Goliath with his slingshot, the composer tackles this behemoth of time armed with his hand calculator, "click track" book, digital metronome, paper, pencils and Excedrin. When he first attacks this monumental task, if he's lucky,

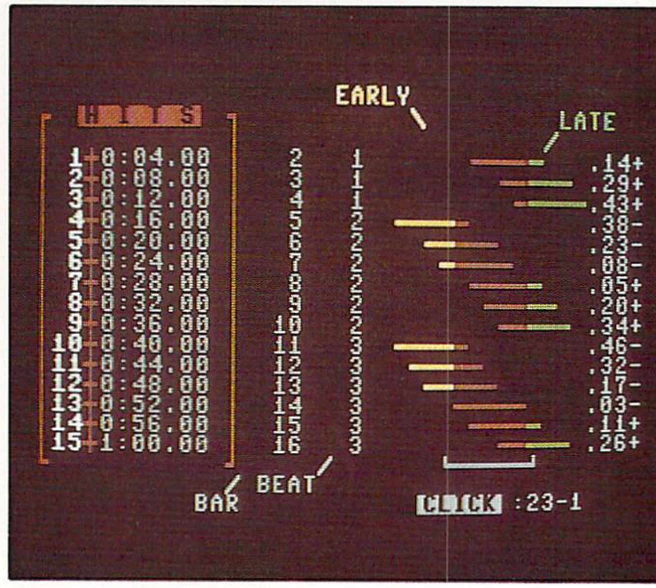
the frames of film, the tempo, the mood and the "hits" he has to make will all line up symmetrically. Does this happen often? Sure. About as often as Mr. Spock giggles.

Charles Bernstein, who composed the score for Columbia's acclaimed motion picture, "Sadat", struggled over one particularly difficult "hit" in the picture. The cue itself was a not overwhelming 30 bars in length—just slightly longer than one verse of "Yankee Doodle." Nevertheless, the timing was irregular, ranging from extremely rapid to abysmally slow in a matter of a beat or two.



Screen photos by Richard Benjamin Grant

The Meter Map is color-coded to help the user easily pick out the meter values for each bar.



The "hit" screen tells you whether, given the current tempo, the music will fall late, early or right on time.

"Just a Commodore" Takes the Industry into

The cover of the April 16, 1984, issue of TIME magazine features MicroSoft top dog, William Gates, smugly balancing a floppy disk on the tip of his finger. The corresponding article, entitled "The Wizard Inside The Machines", offers a capsulized view of where we've been and where we're going to in terms of the "tail that wags the dog": software.

The overall tone of the piece is upbeat and deservedly back slapping. We have indeed come a long way. But the overriding feeling I got was that we have a long trek ahead to the promised land, and when we get there it'll be on the backs of IBM, Apple, Hewlett-Packard and the like.

The article acknowledges that as far as today's software matrix is concerned, we are snared upon a merry-go-round of code words, menus and generally rigid, unforgiving and at times incomprehensible, rules and dictates.

"To reach a truly mass audience," the article states, "software producers will have to achieve a major technological breakthrough. Instead of typing often incomprehensible combinations of symbols, letters, numbers and code words, users should be able to give commands to their computers in plain English."

Further on we are told, "The development of this kind of software is extremely complex, however, and the programs will require extremely powerful machines."

And then they hit us with the long road ahead. "... It will be several years before artificial-intelligence software is widely available."

Wait a minute, fellas. Stop munching on your Apples for a second and take a look up the road a piece. See that guy way up there ahead of you? The one who's *not* driving a Rolls Royce? Yeah, him. How'd he get there and what *is* that thing he's driving anyway? Could it be... is that really ... oh my gosh, it is ... A COMMODORE 64!

And the reigning chieftains, from their magnificent citadels of imperial computer sovereignty, drop their pulldown menus, their overlapping screens and their cute little mouses in a unified gasp of disbelief.

I can empathize with their confusion. Whereas IBM and Apple are unquestionably identified with "serious" computing, as reflected in the way they are promoted and priced, Commodore has, unfortunately, developed the undeserved reputation of somehow being less sophisticated... sort of the not-quite-as-bright little brother. A cute and inexpensive "starter" machine that you can buy at your local Toys-R-Us.

A perfect example of the elitist's attitude toward the Commodore 64 came from the producer of a top-rated television series.

When told about *Auricle: The Film Composer's Time Processor,* he was intrigued by the film production-time and cost-cutting value of the program. When he asked the price of the system and was told, "About one thousand dollars", he pressed, "No, I mean the *whole* system... computer, drives, *Auricle* and all."

He protested that you couldn't even touch a computer with that kind of power for under three to five grand. And since the program used color, there was the color card and the color monitor, not to mention the cost of the program itself. It never occurred to him that the program was written on and developed for a "mere" Commodore 64.

When this news hit him, the "Just-A-Commodore" propaganda he'd heard joyfully took form within his wry comment: "Oh. You mean, the guy couldn't get his hands on an IBM?"

Actually, *Auricle's* creator—one Richard Grant, a lawyer-

Three hours later he finally had it worked out.

Earlier this year, Bernstein got a call from friend and colleague Ron Grant. Ron wanted him to come over and see a new computer program that he and his brother had been working on.

"I don't know anything about computers, Ron."

"You don't have to. Just come take a look."

So he did. And though he was treading the unfamiliar ground of computer country, it was obvious that Ron and Richard's time processing program was unmistakably a major breakthrough for music composers. Bernstein was taken with the possibilities.

He soon learned that the program allows composers to effortlessly adjust the tempo of their scores to fit the visuals. It gives them complete control over every beat within every bar of music. In effect, the program "microscopes" the musical score down to its most elemental component, a single beat, giving composers total creative power over the manipulation of their adversary—time.

Bernstein found the list of "Topicals", or system capabilities, immense and thorough. They include metering, remetering, timings, re-timings, adding hits, tuning forks, clicks, bars, beats, splits, hit cues and tempos. What more could a composer want? Or need?

Well, for one thing, computers are intimidating. To a novice, the language they speak might as well be Sanskrit. Bernstein was concerned about being able to use the program without having to take a six-month night course in computers.

Herein lies one of the *Time Processor's* most unique features. Like the computer aboard the Enterprise, *it speaks English*. If the composer wants to remeter bar ten to 3-8, he types in "retime bar 10 to 3-8". If he wants the program to break down bar 20, he types in "break down bar 20". If he doesn't like the color on the CRT, he types in "recolor crt light blue". If he wants to see the list of Topicals, he types "topicals". *Auricle* is astoundingly user-friendly.

Auricle takes the intimidation out of it.

A second notable feature is the system's *complete lack of menus*. The program will deliver what you want, whenever you want it, regardless of what *it* happens to be doing at the time. Ask and ye shall receive. (I'll have a side of fries please).

Bernstein was spellbound.

Then Ron gave him a pointed demonstration. Using the *Time Processor,* he attacked that dastardly "Sadat" cue that had given Bernstein three hours of agony.

It took Ron all of five minutes. Bingo. Bernstein became a total believer.

Now, of course, Bernstein is a composer not a computerist. As a composer what he saw that evening was the perfect composer's tool. That knocked his socks off. But what I saw when I visited the Grants literally swept me off my feet. It was *Auricle*—the wave of the future, the precursor of Captain Kirk's science-fiction computer—the operating system behind the *Time Processor* program.

It is *Auricle* that provides the most unusual and the most powerful aspects of the *Time Processor. Auricle* ignites the imagination. A prototype for a whole new generation of computer operating systems, *Auricle* could conceivably catapult us into the realm of "where no man has gone before."

Richard Grant explained why they had named their program *Auricle:* "It has to do with the fact that auricle means 'ear'... essentially *the ear of the computer.* What it does is, it takes and breaks the paradox of the screen. Remember Broderick Crawford in 'Highway Patrol'? He would pick up his car microphone and say ten-four! Ten-four! He couldn't talk while the guy on the other end was talking and vice-versa. Well, that's the way

(Continued On Page 18)

the Future

turned-computerist—owns a $25,000 Hewlett-Packard Scientific. And, sure, he has access to an IBM, an Apple or any other computer he might want to fiddle with. But Grant and his composer brother, Ron, are also well aware of the fact that the majority of *users* out there just don't have the kind of bucks to put into one of those systems.

"I wasn't always writing music for television shows," Ron explains. "I wasn't always making a *living* as a musician. Now here we've developed an unprecedented program for orchestrators and composers. Not only does this program make their work easier, but it speeds up the whole process without losing quality. As a matter of fact it potentially improves quality, in that it allows the composer to spend less time shuffling numbers about and more time perfecting the "musicality" of the score. It gives them an edge. But if this program were on an IBM or an Apple, guess what? All the composers out there who haven't yet got their careers going are left behind, because they can't afford to lay out $5,000 on a computer system."

Richard, who wrote the actual code of the *Auricle* program, points out other reasons for developing the *Auricle* system on the Commodore 64. "First of all, it's a wonderful machine, completely functional and *equal in power* to the others. The difference lies in dollar signs. We needed a system that could be replaced instantly during a recording date. Studio time is outrageously expensive. So you've got your IBM out there running the program and something goes wrong. You call up IBM and they say, 'Oh yeah, well we should be able to have someone out there sometime tomorrow'. At $4,000 per hour of studio time? With the Commodore, you can have one on the stage and two in the closet. In all sanity, you just can't do that with an IBM."

But it wasn't just the cost. The Grants found another problem with the bigger, more expensive machines. They couldn't make a "clicking" sound, which is standard on the musicians' click tracks. Instead they produced an irritating "beep". Ron laughed, "I could just see it. The musician's got to sit there and play the violin with this 'beeping' coming through his headphones. No way. He'd go bananas."

The 64's excellent sound capabilities easily produced the all-important "click".

So, *Auricle,* a revolutionary machine-control technique that involves almost every level in the operation of the computer, anarchistic in its disregard of menu-driven standards and totally user-controlled, was created on Commodore equipment.

And IBM, Apple and the rest, all very busily involved in their inner-sanctum competition to make bigger and better menus, more of them, and cute ways to pick them, never even saw Richard and Ron roar past them in their bright and shiny streamlined Commodore 64.

It is an interesting comment on the "upper echelon's" attitude towards us idiot-child users to note that during the developmental stage of *Auricle,* Richard approached Hewlett-Packard with his innovative non-menu concept. The company contritely responded with, "But our users want menus. And we provide our users with what they want."

I think they're going to change that tune. But, too late. The best the other guys can do now is follow the leader of the pack.

The *Auricle* operating system *is* the technological breakthrough that TIME magazine foresaw as an event happening in the future. It's here now.

And who'd of thought it would happen on JUST A COMMODORE?

Kelley M. Essoe

computers work—ten-four."

Auricle takes the ten-four out of it.

The screen is the communications link between the computer and you, the user. It acts as both the computer's mouth and ear—though not concurrently. While the computer is "talking" the screen is a mouth. When it has finished speaking, the screen then becomes the ear, usually prompting you with a "?" or some such signal to inform you that it is now your turn to say something. Pressing the RETURN key is your way of signing out—ten-four—you're done with your turn and the ball is back in the computer's court.

Taking turns, as such, is all well and good. Miss Berilla taught me that in kindergarten. We raise our hand before we speak, we wait our turn at the drinking fountain, and we never, ever interrupt while someone else is speaking.

I always wondered what happened to Miss Berilla. Now I know. She's the one who designed interactive computer protocol.

Personally, I see no need for all this polite behavior with a computer. Particularly since it's so bloody one-sided. When was the last time your computer apologized for rudely interrupting your input with a "syntax error"? Or begged your pardon after a "crash"?

Dear Miss Berilla, with all due respect, sit on it.

The *Auricle* system creates a whole new communications link with your computer. Instead of one channel that must act as both mouth and ear, *Auricle* provides a second dedicated channel of interchange... the ear. The end result of this fundamentally logical, not to mention humane, concept of giving the computer a "face", so to speak, is utterly spectacular.

You can demand of the system virtually any task it is able to perform, at any time, in English, interrupting to your heart's content. All you have to do is ask. The computer will shut up, listen and deliver.

And what a blessing that *Auricle* speaks English... (or French, Spanish, German, even Pig-Latin). The inexperienced user can access the computer without learning a whole new language. As Ron put it, "English you remember. An assemblage of escape-control/C's takes you forever to learn and then you forget a week later."

I couldn't agree more. Let the computer be the one to go to the Berlitz School of Languages.

Put quite simply, it's a matter of who—you or the computer—has to sit and flip through a manual in order to find a specific command so that you both can get on with it. And we know who can flip pages faster.

By referring to her dictionary of "Topicals" in the *Time Processor, Auricle* understands any of the "natural language" commands associated with the functions within a composer's province.

The *Time Processor,* however, is only one application of the *Auricle* operating system. Outside the *Time Processor, Auricle* can contain any number of similar dedicated dictionaries for other types of programs. Or even one large reference that covers a considerable scope of divergent topics.

Menus become an unnecessary and obsolete "tripping stone" of the past. You simply don't need them. With *Auricle,* they'd only get in the way.

What's that, you ask? No menus? NO MENUS AT ALL?? That's right. It's an Emancipation Proclamation. Free the victims of menu tyranny! No more Main Menu. No more choosing what's behind curtain number three. No more thrillingly frustrating attempts at trying to find the sub-menu that will eventually take you to your desired task.

But how will we know how we're supposed to use a program without menus to guide us? Simple. Ask *Auricle.* She'll show you her list of "Topicals" or program functions. And once you are familiar with a certain program's abilities you'll never have to ask again.

Except, of course, when you want to change the semantics to fit your own needs, conventions or personality quirks.

Yep. If you prefer calling a particular function by a name other than what the software author chose to call it, change it to suit yourself. After all, it is *your* software, *you* bought it and *you're* the one who's going to use it. Maybe you'd rather call the "merge" function "mix it up," instead. Maybe you don't like pedantic computer vocabulary and find it easier to deal with "fill-ins" as opposed to "variable blocks". Or perhaps you're a computerese die-hard and long for escape-control/M over simple English. Go ahead, indulge yourself. You can always change it tomorrow. You're the boss.

Richard, a man with a sense of humor, has programmed his computer to call him "Bigshot". "Whadda you want, Bigshot?" They have a wonderful relationship.

Which brings us to the essence of *Auricle.*

We are in the era of "personal computing". But, until now, users have had to adjust themselves to the software. Usher in what Ron calls "personal programs". *Auricle* creates an environment wherein the *program* adjusts to fit the user.

As Richard puts it, "We're not computers... we have a right to be confused! We have a right to not know exactly what we want or in precisely what order we're going to want it!"

Auricle forgives us our innately haphazard thought processes. She permits us the non-linear randomness that gives birth to our creativity. *Auricle* yields to our fits and starts and allows us to be—forgive me, Mr. Spock—human.

On screen, *Auricle* appears as a rectangular window. A funny shape for an ear, but then beauty's only skin deep. The user can slide this window all over the screen or, when it's not needed, have it disappear altogether. Whatever you type within the confines of this window is immediately "heard" by the computer. It's like whispering in its ear. And if you want to get someone's attention... whisper.

Auricle can literally change

New Commodore 64 Word Processor

Cardco's *Write Now*

Write Now is Cardco's new cartridge word processor for the Commodore 64 with built-in 80-column display. Including the features expected in professional word processors that cost much more, this program includes unique and exclusive Cardco features not available on other Commodore 64 software. A few of its features include: easy full-screen editing; interfacing with any printer; special codes transmitting to printers to maintain justification; unlimited recall; an exclusive feature making it simple to replace or find copy; and full-block command enabling the moving or deleting of copy blocks. This program interfaces with Cardco's *Mail Now,* the professional mailing list processor software; *Spell Now,* the 30,000 word dictionary; and *Graph Now,* the full-featured graphics and chart processor.

Circle Reader Service No. 500

the face of computing as we know it. Imagine these capabilities within a database program, or a word processor, spreadsheets or business applications. Picture what it could mean to creating graphics. Consider modem communications, games and adventures. And how about programming utilities?

Richard found that at a certain point in the writing of *Auricle: The Film Composer's Time Processor, Auricle* actually became a major programming and debugging aid.

He discovered that he could ask the computer where a specific module was, such as "Click Cue", and the computer would come back with the proper hex number. While using *Auricle* within a BASIC program he was writing, he could ask "What Is Next F", and the computer would tell him where in the program he had available space—where his next function slot would be. It occurred to him that without much ado, he could literally disassemble the program *while is was actually running.*

"But I haven't even begun to elaborate on that process", says Richard. "The potential is limitless."

Pretty exciting stuff. And with speech synthesizing just around the corner...

Auricle is not yet on the market. But, within the *The Film Composer's Time Processor,* she *has* made her film debut.

On April 5, 1984, at MGM studios in Los Angeles, *Auricle* was involved in a television first. The responsible party was Lance Rubin, a composer for the hit show "Dallas", and one of the first *Time Processor* addicts. At his scoring session for this season's final "Dallas" episode, the variable "click-tracks" from which the orchestra kept time were provided by a Commodore 64 and the Grants' program, *Auricle: The Film Composer's Time Processor.*

So ready or not, here it is, the future... brought to you by *Auricle.*

I'm ready. Beam me up, Scotty.

Kelley Essoe is a freelance writer who lives and works in southern California.

Software Numbering System

The Subcommittee of the American National Standards Committee Z39, an organization for library and information sciences and related publishing practices, has agreed upon a numbering system to be assigned to software for micro and minicomputers.

The assigned numbers will identify "shippable" units, i.e., unique items that can be purchased, shipped and inventoried distinctly from any other. This will facilitate inventory control, ordering, royalty accounting and sales tracking by manufacturers, distributors and retailers. In addition to the number, the standard will recommend and define certain elements of bibliographic data, which will become part of every program description. This will include such variables as the minimum memory/hardware configurations, version numbers and operating systems.

Each number will consist of the following components:

Registrant: The organization or individual who makes the item available for sale. There are some 15,000 potential registrants in existence today and many more are expected to enter the field in the next few years.

Product: The name of the program. There are now 40,000 unique programs available for micro and minicomputers, with the number increasing dramatically on a daily basis.

Delivery Medium: This element is a combination of the physical medium on which the program appears, the operating system(s) required for its operation and any other distinctive information that uniquely identifies one offering of the same program from others available from the same registrant.

Check Digit: This will be a mathematical check digit to verify the preceding numbers.

Telecommunications Brokerage: *Buy and Sell in Minutes*

Fidelity Brokerage Services, located in Boston, has introduced the first national computer-based home brokerage trading service. Customers may enter buy and sell orders on listed and OTC stocks and options, 24 hours a day, using virtually any personal computer equipped with a modem. Investors can also obtain quotes, update their portfolios and review their tax records.

Investors access the service by having their computer dial up a data communications service over local telephone lines. Next, they use their personal security codes to access a database, and then forward their orders to Fidelity. The orders are reviewed by Fidelity registered representatives before being transmitted to the exchanges.

Placing an order takes less than one minute. Market orders can be executed and reported back through the investor's home computer with similar speed. And trades may be settled automatically, through a Fidelity USA account or a Fidelity money market fund.

Phase I offerings permits computerized trading of stocks and options and customer creation of portfolio and tax records. Phase II features, to be implemented through the remainder of 1984, will include trading of mutual funds and direct online access to account balances and positions as they appear in the customer's Fidelity Brokerage account.

Also in Phase II, customers will be able to send electronic mail to their local Fidelity Investor Center and will have online access to Fidelity's asset management account, Fidelity USA. These additional services will be offered to Fidelity Investor's express customers at no extra charge.

Fees include a one-time subscription charge, and timesharing and data communications charges based on usage. Fidelity offers its standard discount brokerage commission on transactions. Stock market quotes are delayed approximately 20 minutes, but up-to-the-minute price quotations are available as an option.

PETSpeed Guide Available

CompuSystems Management announces that the *PETSpeed User Guide* is now available at the retail price of $17.95 including postage. The User Guide covers such subjects as: an introduction to compilers and PETSpeed terminology, operating hints and tips including error codes, treatment of BASIC program codes, PETSpeed and Machine languages, and sample BASIC code programs. For more information, contact Joe Rotello at CompuSystems Management, 4734 East 26th Street, Tucson, Arizona, 85711. He can also be reached by phone at 602-790-6333.

Regional User Group Convention

The Mid-Atlantic Regional Commodore Convention will take place on Saturday and Sunday, July 28 and 29, at the Hershey, Pennsylvania, Convention Center. The convention is sponsored by MARCA (Mid-Atlantic Regional Commodore Association), a confederation of Commodore user groups in the mid-Atlantic states.

Keynote speakers will include Jim Butterfield, Ellen and Jim Strasma and Len Lindsay. Also featured will be original equipment manufacturers and vendors. Call 717-486-3274 for information and reservations.

Computer-Assisted Recruitment

America's largest and most sophisticated corporations have thus far not been able to improve greatly on the old-fashioned corporate recruitment "mating game" despite expenditures for recruitment activities estimated at some $11.5 billion annually. Much of this is spent for recruitment advertising, a notoriously inefficient way to fill a job opening. Employment agencies and search firms provide better targeted services, but are costly and have many limitations.

Computer Assisted Recruitment International, Inc. in Schaumburg, Illinois, provides the obvious answer to the problems of recruiters and job candidates alike. The company can store enormous amounts of information (in this case, profiles of job candidates) in its database, and, in minutes, can screen the entire database to identify a limited number of candidates who most closely conform to the employers' specifications.

Properly set up and maintained, such a system can eliminate virtually all of the inefficiencies of conventional recruiting practices. For example, a company can conduct "silent searches," without tipping its hand and attracting torrents of inappropriate applicants. At the same time, candidates in effect can conduct effortless job searches, with their qualifications and experience constantly available for matching to the needs of potential employers. The capability of computer systems to do this with matchless speed, accuracy and economy is beyond question.

The key to CARI's success is the conversion of candidate resumes into "profiles" wherein the qualifications and experience of the candidates are presented in terms of the criteria established by employers. Candidates are further categorized within specified professional groups, including data processing, engineering, accounting/finance, sales/marketing and human resources.

The CARI system understands that the candidate database must be extensive enough to be of value to corporate recruiters and has set up a system to encourage candidates to supply data. They need only send in their resumes. CARI will edit the data and enter it into the database, free of charge.

In a typical search, the employer-user accesses the desired group, then specifies criteria A, B, C, and D. In a matter of minutes, using virtually any computer terminal or personal or portable computer with dial-up capability, the employer has access to all the profiles in the database meeting each of the specified criteria. If the number of resulting profiles is too large, the user can "fine tune" by adding further criteria—E, F, G, etc.,—until the number of qualifying profiles has been reduced to the desired number. If the number is too small, the original criteria can be relaxed. The user then requests "contact information," permitting the company to get in personal touch with the job prospects.

Unlike some other computer recruiting systems, the CARI system does not require "dedicated" equipment. No special equipment is needed, nor is computer language used. Anyone can be trained to use the system in a very short time. Furthermore—and of utmost importance—the system does not depend on a single central computer. CARI utilizes General Electric Information Services Company's nationwide teleprocessing network; large numbers of users may access the system simultaneously, and at all hours, 365 days a year.

Royal Salute for Commodore

Commodore U.K. has become the first computer company to be given the coveted "By Appointment" designation by the Queen of England. The U.K. subsidiary now joins an elite group that includes such prestigious companies as Bollinger champagne, Cooper's Oxford marmalade, Barbour jackets and Asprey's jewelry.

Any company that regularly supplies goods and services to the royal family is eligible to apply for this highly regarded nod of approval, which is supposed to indicate quality and longevity. Since this award was not bestowed on a native company, however, it was viewed as a blow to the electronics industry of England, since the microcomputer is one of their strongest assets.

The Queen originally received a personal computer from another U.S. company as a personal gift. She is believed to use it to keep track of her horses and their races. The royal family also uses microcomputers in the accounts department and soon plans to begin computerization of the royal menus.

Software Producers Fight Piracy

Piracy has long been a problem for software companies. Estimates on sales lost due to software piracy are not clear. However, it has been approximated to have been close to $500 million in 1983 alone.

In the early years, companies were too small and programmers too individualistic to mount a concentrated attack on copying. But this has changed. Led by a handful of the most successful software publishers, the industry is banding together.

A group called the Software Protection Fund has been formed in conjunction with the 170-member Micro-Computer Software Association, an offshoot of the Association of Data Processing Service Organizations, which plans to attack piracy by lobbying for law changes, legal enforcement, public education and a technical means of protection. Already a suit has been filed by Lotus Development Corporation against a corporate customer accused of copying Lotus' spreadsheet and file-management program.

However, to succeed, the group will have to overcome skepticism, much of it from the industry itself. Some companies have even gone so far as to remove their copy protection entirely, because computer manufacturers who package software with computers believe users need to make backup copies. Many also believe that the problem is already too widespread to adequately control. Given the fact that the home computer user is out to get the best deal, they think the outcome of such a fight would be questionable.

By 1988, sales of software are expected to hit 200 million packages worth $11.7 billion. With software law in a state of flux, many publishers worry that if an action is not taken soon, they may face increasingly complex problems in the future.

The 64 is a Laboratory Super Calculator

by Hugh Doss

Many people think of a medical laboratory computer as some exotic machine that controls all functions of the laboratory, from analyzing test results to billing the patient. While this indeed may be the case in larger laboratories, the advent of small inexpensive computers has made them practical alternatives to handheld calculators or adding machines in smaller labs.

This was my argument to our hospital administrator when I asked to purchase a Commodore 64 for our laboratory. I also stressed that a small computer would increase the "computer literacy" of my laboratory staff, which could be of great benefit in the future if we decided to computerize hospital operations.

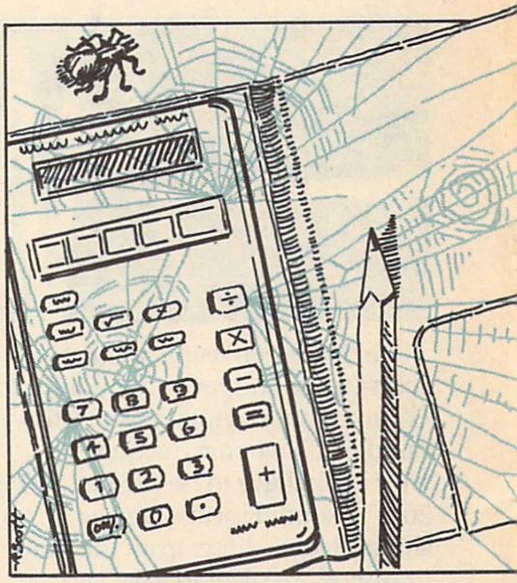
In anticipation of buying the 64, I had previously worked up some short programs on my VIC 20 at home. I was careful not to use any PEEK or POKE instructions in the programs so that they would be compatible with the 64. When we purchased the 64 and a single disk drive in June, 1983, I simply loaded the programs from my cassette tapes and then saved them to disk. With a little reformatting of the screen displays, they were ready to go.

Within a few days, my staff was using the computer almost exclusively for their calculations. In fact, I caught them standing in line to use it even though our other calculators were idle!

Most of our calculations are simple ratio and proportion problems that are easy to do on a far less sophisticated device than a computer. So why does a computer make a good laboratory calculator? Mainly because it reduces the chance of clerical errors. Research has shown that erroneous laboratory reports are due more to errors in calculation of results than to technical errors in test performance. Computers reduce the chance of entering the wrong numbers in a calculation by checking the entries to see if they fit within a reasonable range and by "remembering" constants so that they need not be entered at all.

The alphanumeric screen display is the greatest advantage of all. Individual control ranges for each test, as well as normal and abnormal ranges, can be displayed on the screen when the test is being calculated. This makes it very easy to see if our test is "out of control" and needs to be repeated or not. Abnormal patient results can be flagged as "panic" values and the technician instructed to notify the doctor of the results by phone immediately. This helps insure that no abnormal results are overlooked. Also, notes describing the test procedure itself can be included in the display. This serves as a safety feature by preventing technical errors, especially in the performance of tests that are not done frequently.

In addition, calculation of quality control data such as standard deviations and moving averages has always been a time consuming task. Our computer has greatly shortened the time needed to do these calculations and has allowed us to use more complex formulas than would have been practical



before. This could result in saving several hundred dollars a year—nearly as much as the price of the computer!

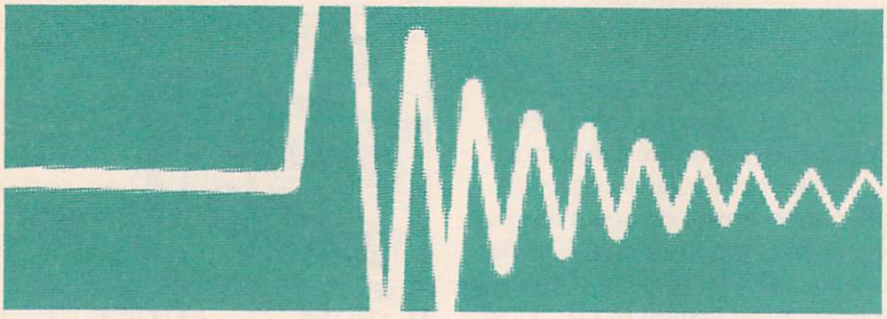
One of the future uses for our computer, when we get a printer, will be as a word processor. We are required by federal law to keep a procedure book, which must be updated each year. This is difficult to do. Laboratory science is presently in a period of rapid change, with new procedures being developed and modifications to existing procedures being made every month. With all our procedures on disk, additions and updates will take only a fraction of the time that they do now.

That, however, is in the future, along with many other possible uses. Our computer's main function will probably be that of "super calculator" for a long time to come. **C**

Hugh Doss is Laboratory Supervisor at Memorial Hospital in Stilwell, Oklahoma.

Commodore 64 Monitors Business Telephones

by Jack Weaver



We have been using, writing software for and recommending Commodore computers since the middle 70's when we acquired what we believe to be the first PET 2001 in south Florida. We had it shipped to us direct from the old Commodore plant in Palo Alto, California, because there were no dealers in south Florida handling Commodore at that time.

In 1981 we arranged for the sale of a Commodore 8032 computer, an 8050 disk drive and a 4022 printer to a large magazine subscription house, Publishers Direct Services in Miami, Florida. Their application was for payroll, (400+ employees per year in two cities), and for spread sheets on their sales and cancellations. Our company wrote the custom software for them, and they have been very pleased with their Commodore equipment.

Mr. Ed Dantuma, the president of the company, told us that he also needed help keeping accurate records of his outgoing WATS line telephone calls. The company was making about 1500 long distance calls per day and had no way to verify the WATS line charges from the phone company. Mr. Dantuma was using a Mitel telephone-switching computer to handle all the switching and routing of calls. He was getting only a sequential printout at the end of each day. The printout was raw data, without any processing, so it really served him no practical purpose. Mr. Dantuma asked us if we could come up with something that would process the data to give him the information to double check the phone company records.

We said yes, we would try.

At that time we were not familiar with the 64, but had heard some great things about it. From what we had read and heard, we figured we should be able to use the 64 in some way to solve his problem. Thanks to some good advice from Lyman Connover of Microbyte Computers in Miami and a couple of phone calls to Jim Butterfield in Canada, we were able to piece together a solution.

The Mitel computer people did not want our 64 computer talking to theirs, so we had to resolve to be strictly a listener. If we sent signals back to their switching computer, we stood a chance of putting it offline, thereby disabling their entire phone system.

The Mitel switcher had an RS-232 printer connected to it, so that is where we decided to tie in to get our data. Lyman designed a simple RS-232 switch box that would switch our 64 on or offline without interfering with the switching computer.

We used a UMI RS-232 interface for the Commodore 64, connected in parallel with the Mitel printer. Data from the switching computer would stream out without any interruption (because we were not allowed to hold up their switching computer) and therefore all data had to be processed and put on disk as quickly as possible.

The 1541 disk drive has a serial connection to the 64. We found it would not transfer data as fast as other Commodore drives. We also found out by our conversation with Jim Butterfield that the RS-232 would hold up a serial device until transmission was done. We then decided to use a Commodore 8050 drive, through a CIE-64 interface. The 8050 drive proved to be sufficiently fast.

All seemed to be going well, until we realized that when the Mitel switcher printer buffer filled up, it would send data faster than BASIC could process it and put it on disk. Naturally the next step was to use a PETSpeed compiler. Now we were able to accept and process data fast enough to get all the raw data to disk. Daily processing of the data would be done at day's end.

The program was designed to accept data all day, from 8:00 a.m. until 9:00 p.m. At the end of the day, one of the employees takes the daily raw data disk, puts it in the other 8050 in the office, which is connected to the 8032 computer,

and uses the 4022 printer to print out the computed data.

The data is broken into many divisions—number of calls from each of 24 phones in the office, number of calls to each of five "bands" in the country and area codes. The exact time in hours/minutes/seconds is computed for each phone in the office, and for each area code in the country. Supervisors are now able to determine the most and least productive stations, area codes, etc., and recommend remedies for the non-productive. And the total long distance time can now be verified against time charged by the telephone company on their bill. In addition, the program also carries forward each daily total and prints out a month-end total of all data.

This system has been running 13 hours a day, five and one half days a week for almost a year, and the only recognizable problem is some data scrambling when there is a power outage. However, this is so insignificant, the company is not even considering a battery backup system.

This type of dependability is something that we and our customers have come to expect from Commodore computers. Obviously the uses for the 64 are limited only by our imagination! We certainly feel that the Commodore 64, properly programmed and applied, is a very effective desktop business computer, with limitless possibilities. C

Jack Weaver is president of Input Systems, Inc., in Homestead, Florida.

A Commodore 64 (left) "listens" to phone calls 13 hours a day and collects data on disk using a Commodore 8050 dual disk drive.

How to Write Your Own Database Ticket

by Donald E. Hassler

If you have read any of my other articles you may have a notion about how a Commodore business computer can help you in all aspects of business. You know how to do your books, how to increase your forecasting and budgeting power and how to get a 24 hour-a-day "secretary" from that Commodore business computer sitting right in your office.

The next big step is to look at database management systems, or DBMS, as I'll refer to them. Just what is a database? A database is a large file of information on a certain subject, organized in special ways. Imagine a filing cabinet with several drawers in it. Each drawer has a label describing the material inside, all on the same subject. That entire drawer is the "file", or database, about that subject. Now inside the drawer you have folders of information, or maybe sheets of paper with information on them. Each folder, or sheet, is the individual "record" in the database. And on each sheet of paper is information filled out, say on a printed form. The individual lines of information are called the "fields".

Another easy way to think of it is to remember the card catalog in your local public library. That file of information is one huge database, describing information about all the books in the library. Each card in the file is a "record" and each category of information on the card is a "field".

Now here is where a computerized DBMS really stands out. In the public library the information has to be cross-referenced by several subjects. For each book you need a separate card for title, author and subject and perhaps even another card for a numerical listing, arranged in order of the standard Dewey Decimal System. But in a DBMS all the cross-referencing is handled automatically, with your Commodore computer doing all the work. The system, if programmed properly, lets you arrange and rearrange records in as many ways as you can think of. It also lets you design reports, or outputs of the information by many different parameters.

A mailing list of customers is a good example of a business use of a DBMS, and perhaps the first important one that a business owner should consider. You've spent thousands of dollars and thousands of hours developing your business and your many satisfied customers. Why not start mining the vein of gold that all those past customers represent and tap some more power of your Commodore computer?

If you would like to design a mailing list of your previous customers, I'll walk you through it step by step. The first step is to think very carefully about what information you need and what you would like to look for in reports and mailings. For example, you will definitely need name, address, city, state and zip. But hold on a minute! Don't rush into the design until you've looked at the particular DBMS program you're going to use. Can you sort on all the fields? Can you change field lengths without losing all the data? What about calculations on output? How are the labels prepared? How wide and deep can they be? You'd better have all the answers to these questions before you go barging ahead.

Here are some suggestions to consider:

1. Have separate fields for first and last names.

2. Consider a field for "Mr. & Mrs.", "Dr.", etc.

3. Leave a ten-character field for zip (remember?)

4. Don't forget the phone number, with area code.

5. Use two fields for the name if any of your customers are businesses, which means you need the person's name and the business name.

6. Add as much information as you would like about that customer. Some things you may need are salesperson's number, invoice number, date of sale, type of merchandise purchased, store or plant location purchased from, personal information such as birthday, spouse's name, etc., dollar value of purchase, type of customer, season of merchandise, etc., etc.

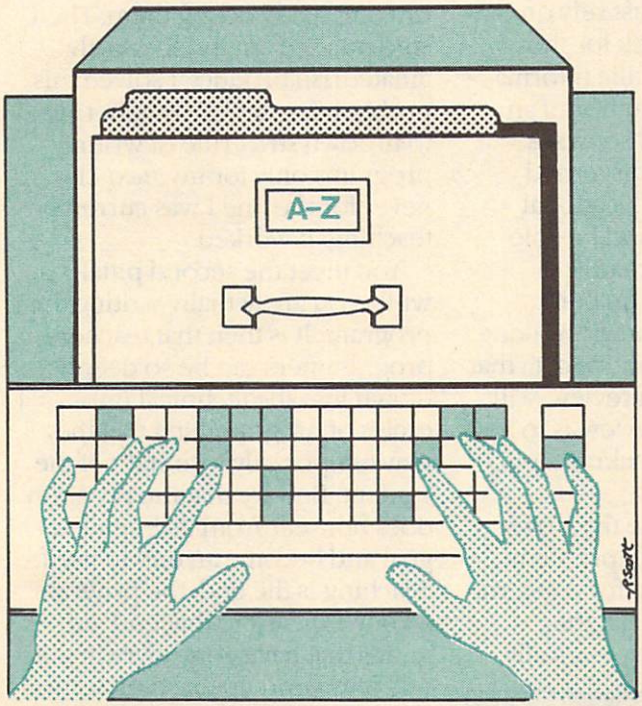
It is possible to have information you'll never need. That will just clutter up the files and use up space. But you must plan this database because once you start, fixes will be very hard. When considering various DBMS's be sure that you'll have enough space in your Commodore computer to get the whole database on one disk. It is very difficult to do sorts and changes with parts of the file in different places. The *Commodore Software Encyclo-*

pedia is a good source of information on the various database systems available.

Here is how the finished form for entering each record might look. Remember, this mailing list is a database, and you can use it many different ways depending on the power of your particular DBMS. Each field is named and the number of characters in each field is shown in parenthesis after the entry line.

Sample Record

Title*******(7) FName************(12)
LName************(12)
Co Name***************(15)
Phone************(12)
Address***************(15)
City************(12) State**(2)
Zip**********(10) Date******(6)
Invoice******(6)
Salesperson******(6) $ Amt***** **(8)
Merch**********(10)
Birthday******(6) etc. etc.



Now a couple of last minute thoughts. Sort your mailing list by zip code first (primary sort) and then by last name. Make sure that when you enter new names, they are sorted the same way. If you have changes, delete the entire record in the file and enter it again by the new information. That will avoid the possibility of duplications and other problems. And after you have used your DBMS program for a mailing list you'll be ready for some *really big* adventures. Next, you'll be setting up a complete inventory system in your DBMS, using your Commodore computer. C

Donald Hassler is president of Fidelity Management Systems in Phoenix, Arizona.

Evaluating Educational Software

by M. W. Caprio

Everything that we teachers do in our classrooms has an impact on our students—the clothes that we wear, our punctuality or lack of it and the audio-visual aids and dittoed material that we select. Who we are and how we teach is a big part of what we teach. So if we decide to use computers, that decision also gives the students a message. Just what that message will be depends a great deal on the software that we select.

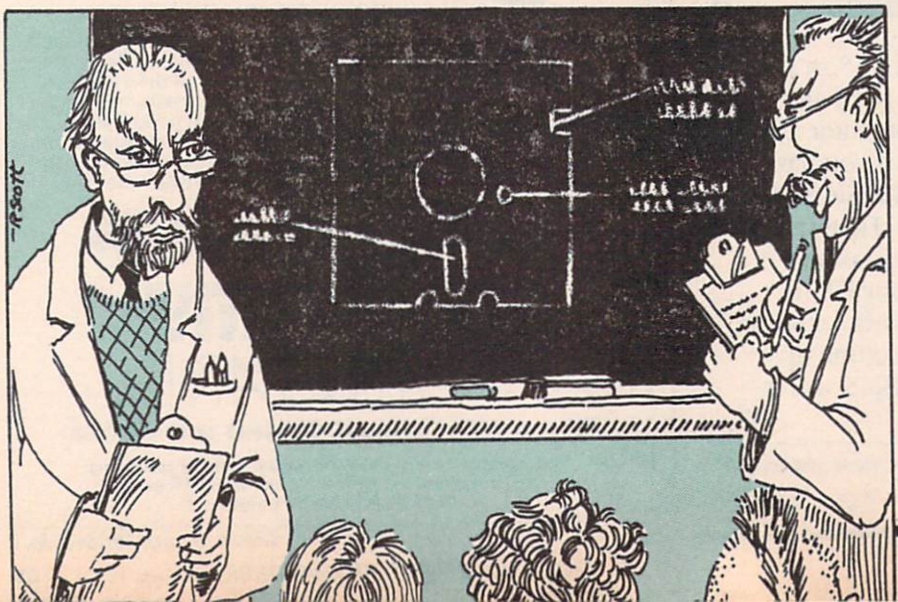
It is imperative, in this age of computers, that teachers extend their techniques in education to include educational software. But, as always, the evaluation of any published teaching material is more valid if that critique grows out of a broad-based philosophical stance. This helps to put the evaluated items into a larger, more significant context. Any one audio-visual aid, textbook or computer program is but a single unit in the architecture of a total teaching effort. Its usefulness depends upon the contribution that it makes to strengthening that total effort. So if software is at cross purposes to your educational philosophy, it will weaken, rather than support, your teaching. That piece of software may be all right for someone else, but it might only undermine your personal effectiveness in the classroom.

When you are looking for software, then, if you must rely on software reviews, look for those that provide the specific information you need. The author of an effective educational software review should have discerned the programmer's educational philosophy, and should be able to describe it to the reader in relation to the program being reviewed. That information, along with the other descriptive data that is ordinarily part of a review, will be necessary if the review is to be of any real value to making a purchasing decision.

Teachers who write their own software have less of a problem, but there is still room for concern. Presumably the material they create will, almost automatically, fall safely within their personal philosophical boundaries, but two major pitfalls remain. First, there is a tendency for enthusiastic teachers, usually those who are new to programming, to be so anxious to get their programs into the classroom that they may never put the finishing touches on them or completely debug them. Their students see an unnecessarily amateurish product. I solved this problem for myself by adhering to my own strict rule of writing programs only for my *next* class, never for the one I was currently teaching. It worked.

You meet the second pitfall while you are actually writing the program. It is then that teacher-programmers can be so deeply drawn into the technical intricacies of programming that they may temporarily lose sight of the student. Beware that the program does not veer from its intended goal and become an end in itself. Teaching is the end; the program is only a means to that end. Programs that have gone awry in this way may wind up teaching something altogether different from what the author intended.

This is difficult to guard against. However, I find that it helps to write down an outline for the program that I intend to develop and to make a list of the subroutines that I think I will be needing. That list is inevitably added to once I start programming. Without question, though, the most important step that helps to keep my programs on target is to methodically review my general thoughts on educational software before writ-



ing a single line and then again about midway through the project.

Those "thoughts" are the guidelines that make the programs effective teaching tools. They are strongly linked to my educational philosophy and they embody time-tested educational values that are conservative enough to be shared by most classroom teachers. I am presenting some of them here as a starting point for beginners in educational programming. You will not agree with all of them and you will certainly want to add a few points of your own. My goal here is only to provide a few handholds for those who are beginning their climb up the educational software mountain and need to develop a method for critically evaluating their own work or the work of commercial programmers.

Let me begin by stating the obvious: computers should be used for what they can do best. I once began writing a program that employed computer graphics showing anatomical structures. Only after many hours of work at the keyboard did I realize that I was working in the wrong medium. The graphics were fine—for a computer—but a child with a crayon could have made better drawings. Strange as it may seem at a time when so many are heaping so much praise on these machines, computers can sometimes be the wrong tool for the job.

When a computer is being used for something it was never meant to do, the quality of the program may still be very good, but its value as a teaching aid is bound to be low. Notwithstanding the truly elegant programming that went into that anatomy example, a much better teaching aid could have been more easily produced with nothing more than a pencil and a few ditto masters.

If they wish to demand excellence from their students, teachers must first set an example of excellence in what they present to those students. Working with the wrong tool is guaranteed to produce less than an excellent product. Computers are not the only teaching tools available to teachers, so it is always fair to ask whether a computer is the appropriate method for accomplishing a particular educational goal.

The first question, then, in critiquing a piece of software is: Can its instructional point be better made in a different medium? If so, consider using the right tool for the job.

However, if your judgment is that a computer is up to the task, then you must evaluate the quality of the software. You will want to know if the program addresses a single concept or if it is too ambitious to be an effective lesson. This can be a tricky decision; it depends on the application you have in mind. For example, a program that randomly mixes addition and subtraction facts would be too confusing if you needed it to teach only one of these topics. But if you wanted something that stressed paying attention to the plus and minus signs, then that same program would fulfill a single concept.

Given that a single-concept program is apt to be a more effective one, we must ask whether or not the concept that it is trying to teach is clear to the learner. This is closely tied to a basic assumption about students that nearly every teacher holds to be true. That assumption is that, in general, students want to learn. It logically follows that if you tell students just what it is that you expect them to learn, they will try to learn it. To that end a good program must make its objectives clear to the students from its very beginning, and it must not digress from that goal as it runs.

Next, when you are evaluating software take a close look at what happens after the student inputs an answer. How the correct answers are congratulated is important. Learning is supposed to be the reward of study. Learning is growth; it is exciting and self-satisfying, but some software rewards learning in less than appropriate terms. For instance, when an "educational" program slyly couches learning in an arcade game format, it is telling the student that, among other things, (1) the teacher does not believe that learning is its own reward and (2) the teacher suspects that the subject may be too boring to hold the student's attention. In one broad stroke it insults the teacher, the subject matter and the student. Programs that do this use the feeble efforts of a machine to try to motivate learning. But motivation comes from the mind; it is very personal and very human. A computer is the wrong tool for this job.

How the program handles wrong answers is equally important. If a program allows the students to guess again and again, they may soon begin to believe that making carefully considered responses that are correct on the first try are unimportant. Students are so visually oriented and so very receptive while they are sitting at that monitor, you must be extremely careful that the software does not insidiously deliver a destructive message to them. The computer is a powerful teaching tool, but you must be its conscience.

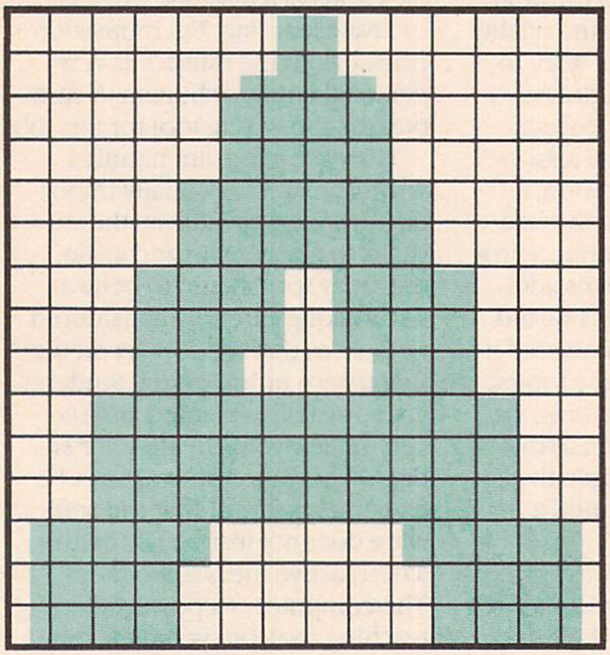
I have not talked about the grade level, vocabulary or grammar of the programs. (Is it all right to only use capital letters?) Nor have I discussed the use and misuse of humor in them. I'll leave these to you. My intent has only been to sharpen your critical eye for educational software, not to write a definitive treatise on the topic. The real work lies before all of us. We have, in computers, a formidable teaching tool. However, its lessons can carry subtle messages that the programmers never intended. The enormous power of the medium demands that teachers become astute evaluators and severe critics of educational programs. C

M. W. Caprio is professor of biology at Community College in Stony Brook, New York.

Grade Master 64

For the Commodore 64

by Rick Jeandell



"Grade Master 64" is a program I wrote to help my teachers keep track of their grades. Up to 40 students' grades can be stored at once. The grades can be averaged, added to, printed out on a printer and saved on tape.

"Grade Master 64" is run by a network of menus. Because of space limitations, two separate menus were needed. The first menu consists mainly of routines not used during the entering of grades, such as the SAVE, LOAD and PRINT options. The second menu contains the options used while entering information. Below is a list of what each option does and how it is related to the others.

Load Grades: Recalls previously SAVEd classes' names, grades and averages. When the file you want to load is found, the message "file open" appears briefly on the screen. Once the variables are loaded, you are returned to the first menu.

File Grades: Saves all students' names, grades and averages entered since you loaded the program or during a NEW CLASS option.

Print Grades: Prints out all necessary information to your printer, such as class name, marking period, semester, period number, students' full names, their average and their letter grade. All latest grades must be averaged before selecting this option because the grades are not averaged as they are entered for each student. Before printing, "Grade Master 64" will check with you to make sure the grades have been averaged. Also, if there are no students in memory, nothing will be printed.

New Class: Used to start a new marking period, another period or just a new class. All variables in memory are cleared. While entering students, if you happened to have made a mistake on the previous name, just hit the English pound sign (£). You can now retype the name. You can keep hitting the £ key all the way back to the beginning of the class, but the names skipped over to get to the name are erased. When all students are entered, you have the option of having the class alphabetized or not. A bubble sort is used to sort the list, so a class of forty students will take a little time.

Second Menu: Puts the second menu on the screen. No choices from the first menu can be chosen while on the second menu.

Extra Students: Used when you already have a class and one or more new students join the class. Each extra student starts out with a 100 average unless you change that in line 1191. After they are entered, those students are now part of the class. If you choose, they will also be inserted in alphabetical order or they can be left at the end of the list.

More Grades: Selected when a new assignment needs to be recorded into "Grade Master 64." The date of the assignment will be asked first. If the letter "c" is typed in instead of the date, you can correct a student's grade. When correcting a student's grade, either the date or name of the assignment is needed to correct it, but make sure the corrected grade is re-averaged before printing. If nothing has to be corrected, enter a date using six digits with two slashes (e.g., 07/01/83). "Grade Master" only checks for the length, so be sure that it is typed correctly. Next, the name of the assignment is required. This is only used for reference purposes and cor-

recting grades. Finally, the total number of points possible on the assignment will be entered. The grades are figured on a point basis—the total number of points the student received divided by the total number of points accumulated over the marking period. After this, all students' last names will be displayed and the grades they received on the assignment are typed in. Nothing higher than the total possible points on the assignment will be accepted. When all students' grades have been entered, the option for another assignment or return to menu is displayed. If another assignment is chosen, you will be taken back to where the date is asked for.

See Grades: This option displays, one at a time, the grades students received on individual assignments. This is helpful to check that the right grades were entered.

Average Grades: This is required before printing out on the printer because grades are not averaged as they are entered. Choose this option at the end of the marking period for report cards or just to see how the class is doing. Only a number average appears on the screen but both a number and a letter average appear on the printout. Ninety or above is an A, between 80 and 90 is a B, between 70 and 80 is a C, between 60 and 70 is a D, anything lower is an F. This scale can be changed in lines 2551-2555.

First Menu: Displays options of the first menu. None of the second menu options can be used while the first menu is displayed.

Instead of the flashing cursor, a flashing line prompts input. I used this subroutine to keep out the cursor controls and clear the screen. If you hit the wrong key while on the menu, don't worry. Just hit the back arrow and you will be returned to the first menu. But *beware,* any time this key is hit you are returned to the menu, even if you are in the middle of typing in a class. I used the back arrow because it was out of the way. This key can easily be changed by changing line 179. If you just want to quit, hit q while on the first menu. This doesn't save anything, so if you hit q by accident, type "GOTO 1050" and you'll be back at the first menu.

Since the program is too long to publish here, just mail a blank tape or disk with $2.00 and an SASE, and I'll be happy to make copies. Mail to:

Rick Jeandell
430 Smiths Mill Rd.
Newark, DE 19711

C

Software Distributors and the "Best Effort" Clause

by Herbert Swartz

Before you try to market that great software package you just wrote, read this.

LICENSEE agrees to use its best efforts to market and promote the SYSTEMS during the term of this License Agreement, and to exert its best efforts to sublicense the SYSTEMS to customers as herein provided.

For years, this "best efforts" clause, though frequently a part of distribution agreements of all types, has been treated as a legal stepchild. Rarely invoked and even more rarely litigated, it has been the subject of but one major judicial interpretation during the entire twentieth century.

Yet suddenly the doctrine is front-and-center in computer law. As Esther Dyson, president of Rosen Research in New York and publisher of the computer newsletter *RELease 1.0,* notes, "It's one more thing to consider if you decide to be a software distributor." In turn, it is one more equitable and realistic opportunity for computer manufacturers and software authors to have their works treated properly in the marketplace by distributors and publishers.

The relevant chain of events began in 1979. Information Access Systems (IAS), a small New Jersey developer of DEC-compatible software, licensed Heath Company of Michigan, a subsidiary of Zenith Radio Corporation, as its distributor. To put it kindly, Heath went about fulfilling the contract with lassitude.

No patsy, and quite riled up, IAS proceeded with zeal. It retained Peter Brown, partner in the New York firm of Brown, Raysman & Millstein, a celebrated writer and lecturer as well as practitioner of computer law, to bring suit. The contract contained the "best efforts" clause quoted above. When the legal dust settled in a Grand Rapids federal court, the jury awarded IAS damages of $1.3 million.

Though motions to overturn the verdict and reduce the amount of damages, and of course the obligatory appeal, remained unsettled at press time, if IAS should continue to prevail, the judgement, with interest, may reach close to $2 million. Comments professor Joseph Perillo of Fordham Law School, co-author with his Fordham colleague, professor John Calamari, of The *Law of Contracts,* "The case officially makes computers like other products in our society."

The issue then is to determine what that entails.

In the landmark "best-efforts" case, *Bloor v. Falstaff Brewing,* decided by the United States Court of Appeals for the Second Circuit in 1979, the court held that a distributor must "merchandise" the licensed products "in good faith and *to the extent of its own total capabilities.*" (Emphasis added.)

"Total capabilities", be it noted, is a far greater duty than a distributor might consider appropriate for its own products. For with its own products, the goal is profits, of which high volume is only part. Given direct and indirect cost of sales, "high volume" and "profitability" are not synonymous. So, the Second Circuit said, the "licensee may promote, continue and discontinue its own products as it wills, subject to its duty to shareholders."

But in the instance of the licensed product, the need is volume as high as possible, for the licensor receives a royalty on each sale; each gross royalty dollar received is a net dollar. The more sales, the more royalties, the more net, and as the court concluded, that is "the contractual obligation" to which a distributor is bound.

The court did not suggest that a distributor must spend itself into insolvency to fulfill the "best efforts" clause. Rather, it concluded that "total capability" is much more inclusive than "financial ability". It includes "the marketing expertise and experience attributable to the 'average, prudent, comparable' " distributor. And therefore, "the extent of its own total capabilities" is the standard a distributor must fulfill.

Examples abound of publishers who contracted to distribute a book, or motion picture companies who took on a film and other distributors of all stripe who simply put the licensed work on the shelf. Software authors and computer manufacturers are not immune from this happenstance. It is no way to run a computer industry, and now, given the IAS victory, such a prospect has been largely diminished.

Obviously, getting a distributor to agree to a "best

efforts" clause is no light task, and the IAS case will surely increase the difficulty. Yet Lance Rose, general counsel for John Brockman Associates of New York, a leading agent for software developers, says some distributors are quite willing to include the clause. Others, however, are adamantly opposed. Thus negotiation is the key.

Rose considers inclusion of the provision so vital that he suggests foregoing "a few tangibles"—for example, the size of the advance or the royalty percentage—in exchange. The lesson, manifestly, is that a higher percentage of nothing is nothing, while a smaller percentage of a duty to sell guarantees some dollars, maybe more in the long run. " 'Best efforts,' " says attorney Esther Roditti Schacter of Schacter & Froling in New York, "doesn't mean the distributor has to succeed. It does mean that he can't just sit back and do nothing. With 'best efforts,' distributors have something legitimately to be worried about."

Worried, indeed, and not just those computer or software distributors who choose to do nothing. With "average, prudent and comparable" as the standard of necessary performance, no cutting of distributive corners will suffice, either, because in the computer industry the "average" standard is high. "I go home at night and sit down to read the *New York Times*," says Schacter, "and all I can find to read are the computer ads."

Hyperbole aside, the marketing obligation incumbent on today's computer or software distributor is very complex. Says Fred Gardner, executive editor at *Marketing & Media Decisions,* "Marketing people aren't clear on where to go. There is more than merely the computer [trade magazines]. Some people are starting to go into the women's magazines. That way they'll reach the parents, the adults who actually buy the software. They're seen as the best prospects."

Further, Gardner continues, there is the question of internal procedure for correct computer distribution. "You need a separate copywriter department to handle what you license as distinguished from your own products. To do things properly, you also need separate media units."

In short, "best efforts" is "still a vague obligation," for *both* sides of the license agreement, as Rose points out. A vague threat to distributors, yes—but too vague, as well, to be the sole and ultimate solution for software authors or manufacturers. For the latter, "best efforts" is something to be sought, but not to the exclusion of "quantifying the distributor's obligation," in the words of attorney Mark Gordon of Gordon & Glickson in Chicago.

In other words, placing a figure on advertising dollars to be spent, people to be utilized, the marketing breakdown, the hours expended and whether or not reps should be hired—all these, says Gordon, "are valuable and irreplaceable in their own right", in addition to any "best efforts" clause.

If each criterion can be put in numbers, says Rose, you can expect better performance. "The contract becomes a reference tool for both parties while the project is still active. People can check consistently on what they are supposed to do, and what is supposed to be done by the other parties."

Such benefits, Gordon allows, extend especially to contracts granting exclusive distribution rights. They provide the shibboleths by which a licensor can feel free to change the "exclusivity to nonexclusivity, or terminate the contract altogether. They are valuable spurs to prod a distributor into better performance."

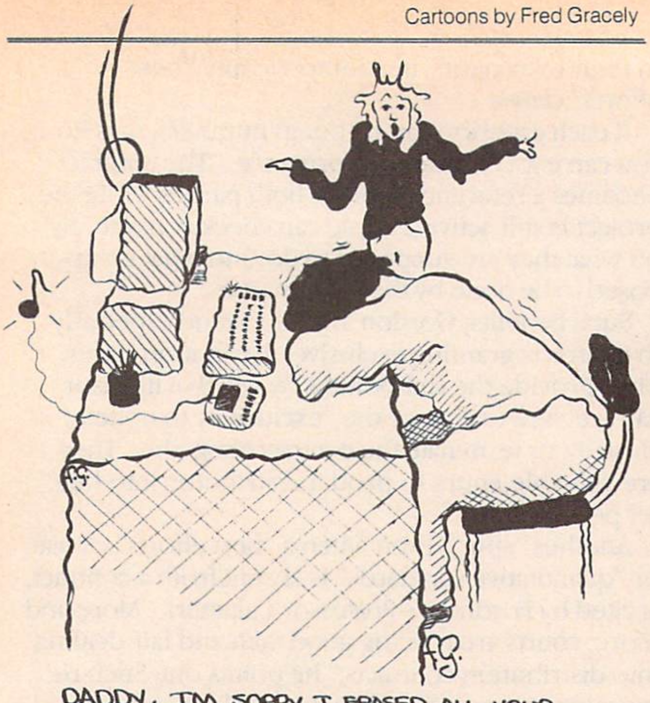
Another "spur," even when a "best efforts" clause or "quantitative standards" is absent from a contract, is cited by Fordham's Professor Calamari. "More and more, courts are reading good faith and fair dealing into distribution contracts," he points out. Such requirements may not rise to the level of "best efforts" or quantitative numbers themselves, but they, too, mark as past the days of shelving a product or giving it the once-over-lightly with impugnity.

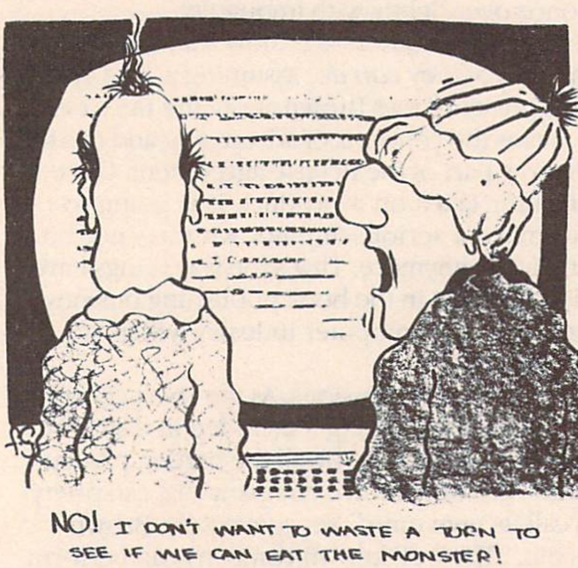
Computer distributors are "now subject to the maxim of what they *can do,* assuming a 'best efforts' clause is present," says Brown about the IAS case. "This means the concepts of advertising and marketing are now part of the license agreement. Once a distributor takes on a product, it is going to have to make a serious attempt with it—not be free and loose anymore. That's the way things have traditionally been in the book publishing business, but let's hope the computer industry will go a different route."

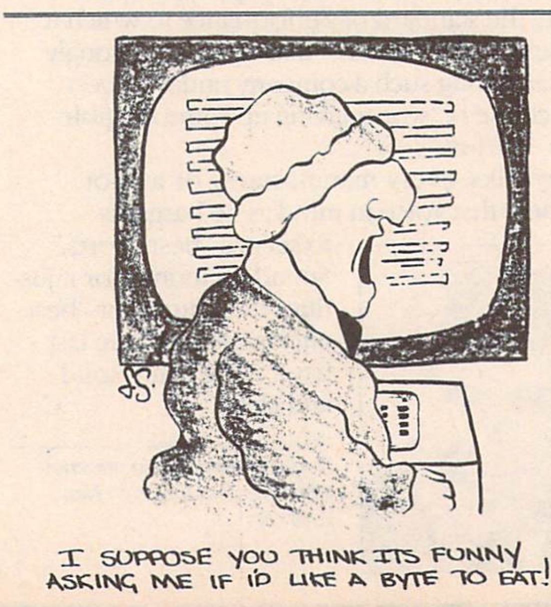
But Brown has no illusions. As the "best efforts" doctrine burgeons, having it included in distribution contracts is going to be all the harder, particularly with "the well known company that can pretty much call its own tune." Yet conversely, Brown points out, "The well known company can perform very well; the standard of performance to which it will be held is very high." For that reason, he strongly urges that getting such a company under a "best efforts" clause is "worth giving up some tangible rewards up front."

Money talks. Every manufacturer or author should bear that voice in mind as he bargains to secure "best efforts." Spending money for it justifies the return, for "best efforts", too, at long last has a voice that is loud and clear. **C**

Herbert Swartz is an attorney who lives and works in New York City.







Editor's Notes

(Continued From Page 12)

works. Its headquarters—and mainframe computer—are in Columbus, Ohio. But when you subscribe you don't have to call Columbus to access information. Rather, you get a local phone number that then connects you directly to the main computer without costing you long distance rates. All you pay for is "connect time"—at a certain rate per minute. Like phone rates, connect-time rates vary according to the time of day. (However, although large networks like CompuServe generally work this way, many smaller services and bulletin boards have only one phone number, which, if you're not in the area, *will* mean long distance charges.)

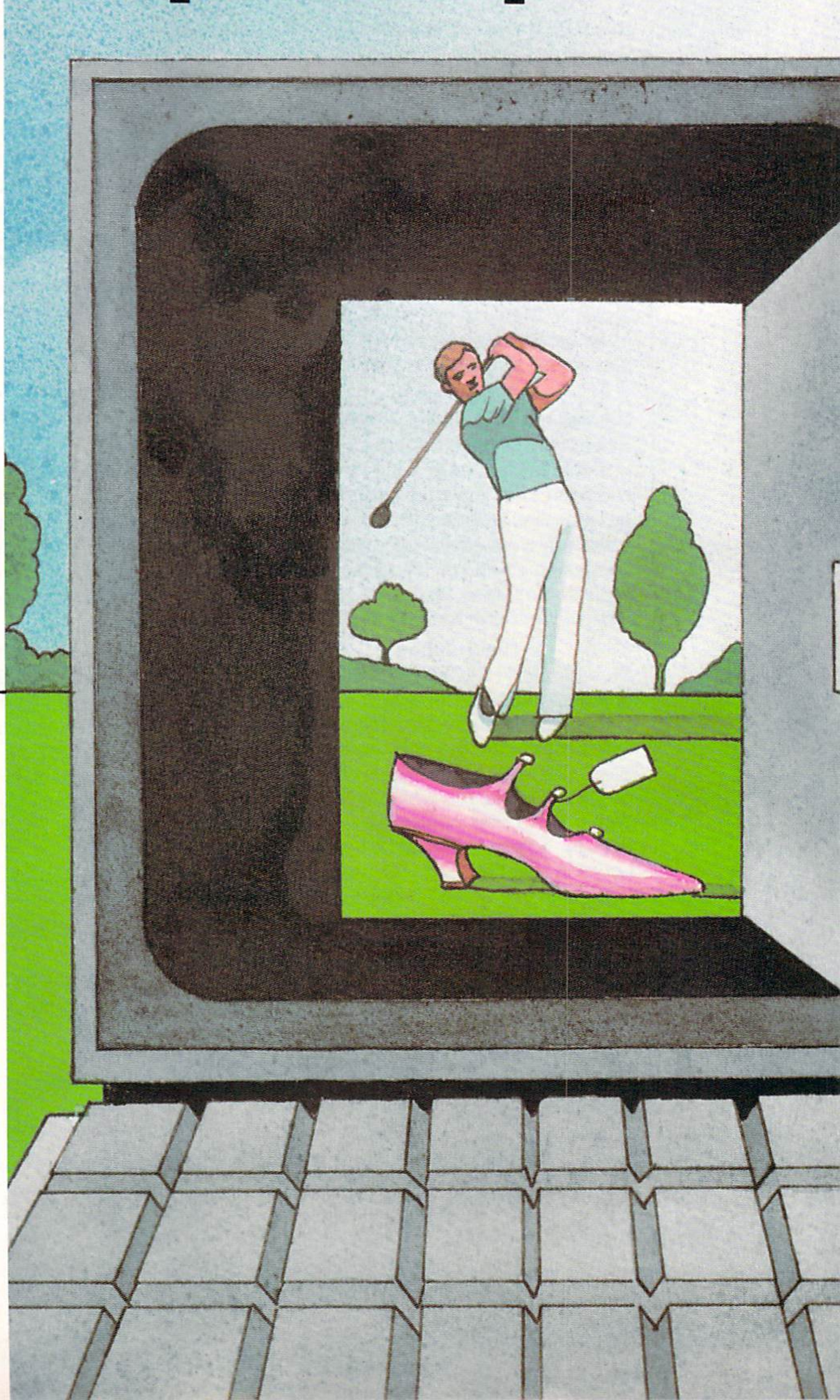
Okay, so you're connected to CompuServe. Now the fun begins. Try browsing through the many levels of menus (lists of what's available) to find out what's there. Under "Home Services", for instance, maybe you'll stumble over the "Cook's Underground". Or maybe you'd like to do some banking—or shop the "Electronic Mall". Get the latest stock quotations in the "Business and Financial" section. Find out who's buying and selling diamonds in the jeweler's SIG (Special Interest Group). Or plug into the literary SIG. Or, better yet, go to the Commodore Information Network in the "Personal Computing" section.

After a while you learn the "page numbers" of the areas that interest you, so you can go directly to those areas, bypassing the menu structure. Or, if you need information on a specific topic and you don't know where to find it, you can go to the index, type in the topic name, and get a list of all the places on the system that deal with that topic.

If you're new to telecommunications, I hope this gives you a vague idea of what it's all about. For more detail, however, you'll have to turn to our features section in the middle of the book. C

—Diane LeBold
Editor

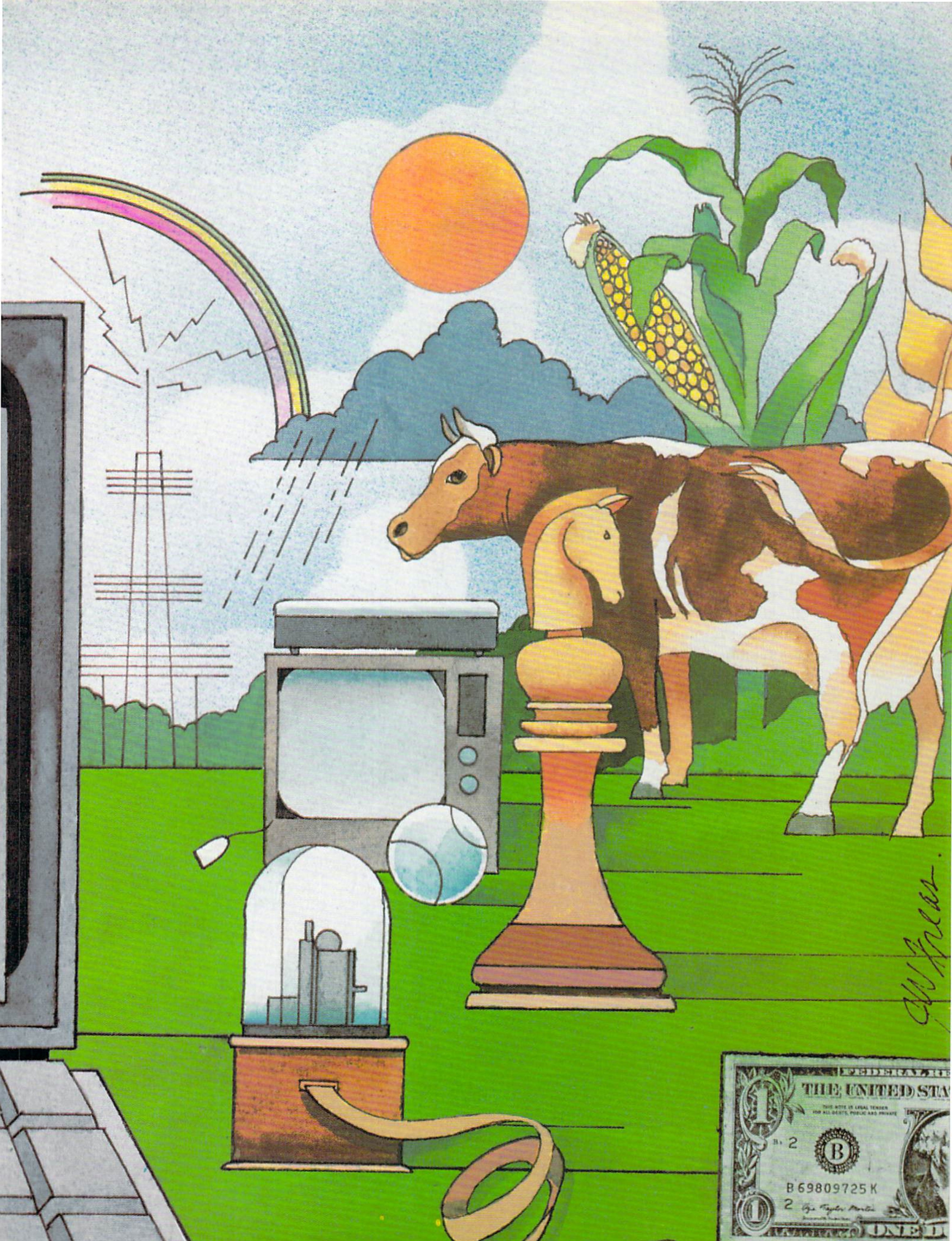
A New World Opens Up



Telecommunications can be a great world to explore! It can, or soon will, deliver everything from entertainment to technical help to recreation. New uses and applications keep springing up each day. It is even rumored that one of the major reasons many people buy a microcomputer is to telecommunicate. If you're new to telecommunications, here's an overview to get you started.

By Barbara Karpinski
Commodore Telecommunications
Product Manager

Taking the plunge into telecommunications can be overwhelming as well as exciting, however, and can sometimes discourage even an enthusiastic computerist. But most confusion can be cleared up if you take in the entire picture instead of focusing on its individual parts. Once we know what telecommunications involves, what's available and what we need to do to get at it, the pieces start falling into place. What I hope to accomplish in this article is to provide a basic understanding of telecommunications—a sort of foundation on which to fall back when you're faced with decisions concerning telecommunications products and services.



Illustrator—Jack Freas

The Basics

The *Computer Dictionary* defines telecommunications as the transmission or reception of signals, writing, sounds or intelligence of any nature by wire, radio, light beam or any other electromagnetic means. In the sense that we will be using it, telecommunications is the communication of two or more computer terminals set up in various combinations.

These are the basic combinations for hookup:

1. Micro to micro
- A. In straight communication
- B. One set up as a bulletin board

2. Micro to mainframe
- A. As a terminal for a mainframe, with direct access (e.g., a school computer or business computer with isolated terminals)
- B. Information services with indirect access

In order to be able to turn your computer into a communicating terminal you will need the following essential equipment.

1. Microcomputer:
A digital computing system whose main processing blocks are made of semiconductor integrated circuits. It does not become a communicating terminal until you load in your terminal software and connect your modem.

What's Available and How To Get To It

Now that you have an idea of the equipment you will need to get started and the basic hookup combinations, here's a partial list of what's available in each category. These are just examples of the kinds of telecommunicating you can do, and are by no means the final word, but they will give you an idea of how to start.

Micro to Micro

1. Commodore 64 owner communicates with another 64 owner.

2. Bulletin boards (BBS) can be defined as electronic corkboards where messages can be placed and received. Some BBS's also allow you to "talk" with the system operator (sysop) or with other members in real time and to download/upload public domain programs. Most local BBS's are free to access. The only cost incurred is the telephone call itself.

On this page is a list of bulletin boards that run on the 64 and/or are set up for the VIC/64, arranged alphabetically by state.

3. A novel way to communicate between micros is being pioneered by Mad River Video of Warren, Vermont. Using this system, FM radio stations can

Telecommunication Bulletin Board

United States

California:
Magnetic Fantasies
sysop: Claude Plum
(213) 388-5198

Illinois:
Commodore Chicago
(312) 397-0871

Chicago/24 hours
sysop: Keith Peterson
2246 North Palmer Drive
Schaumburg
(312) 397-0075 (voice)

Illinois Video King
Chicago/24 hours
sysop: Gene Alper
(312) 674-6502

Indiana:
AVC-Commline
Indianapolis/24 hours
sysop: Tim Renshaw
(317) 255-5435

Louisiana:
NHUG
sysop: Gary Pang
(504) 467-9897

Dante's Inferno
(504) 392-4156

YAT's
sysop: Glen Martin
(504) 279-3832

NED/SIG
sysop: Doctor Charles C
(504) 737-8173

Dungeon
sysop: Mike Perry
(504) 245-8920

Massachusetts:
VIC BBS
Natick
(617) 478-4164

Missouri:
#1 Bulletin Board
BBS # (314) 388-1293
help # (314) 869-2222
contact: Steven Glass
24 hour BBS with a 30 minute time limit contains more than 50 programs for the 64

Kansas City PET User Group
c/o Blue Ridge Blvd.
Kansas City
(816) 356-2382 (sysop)
(816) 356-6502 (voice)

Commodore Communications
(314) 625-4576

St. Louis/24 hrs.
sysop: Tony Ott
Commodore Communications
633 Bent Oak Drive
Lake St. Louis
(314) 625-2701 (voice)

New Jersey:
64 BBS
(609) 667-9659
hours of operation: 6PM-6AM weekdays
24 hours on weekend

2. Modem:
A device that converts data that the computer can understand (usually as eight-bit bytes) to a form compatible with transmission lines (usually one bit at a time). The rate at which this transfer is made can be at 300 baud (bits per second) or 1200 baud. Most commercial services such as CompuServe and Dow Jones charge extra for 1200 baud, and whether or not you need a faster modem depends on your application. Also some modems offer extras such as automatic answer/dialing and modem programming capabilities.

3. Terminal software:
Terminal software will convert your computer into a terminal, ready to telecommunicate. It can come in either disk, cartridge or cassette form and with different levels of sophistication. A basic terminal program has the ability to display information (send and receive) on your monitor or television set. This is sometimes referred to as a dumb terminal program. A smart or sophisticated terminal program allows you to manipulate (to some degree) incoming/outgoing information and to save this information on a hard copy medium such as a disk drive or printer.

4. Modular telephone:
Of course even if you have your modem connected and your terminal program loaded, you still need a phone line to connect to the other computer. A modular phone is one that disconnects either at the base of the phone or at the handset of the phone (cord unclips).

In addition, you can use the following optional equipment in telecommunications.

1. Disk drive: For hard copy and/or loading in your terminal program if it is saved on a floppy diskette.

2. Printer: For a hard copy of information that you've received on a telecommunications system.

64 BBS/24 hrs.
(609) 667-8340

Ohio:
Ohio Valley
Marietta/24 hours
(614) 423-4422

Texas:
RPCC
Dallas
sysop: Don Lambert
(214) 996-7994
hours of operation: 5PM-8AM

Washington:
NWCUG
Edmunds (Seattle Area)/24 hours
sysop: Dean Johnson
(206) 743-6021

Wisconsin:
SEWPUG
Racine/24 hrs.
sysop: Tim Tremmel
(414) 554-9520

SE Wisconsin PET User Group
3614 Sovereign Drive
Racine
(414) 554-0156 (voice)

Wyoming:
SE Wyoming CU
Cheyenne/24 hrs.
sysop: Roger Kelsar
(307) 637-6045

SE Wyoming CU
c/o Computer Concepts
1104 Logan Avenue
Cheyenne
(307) 632-9132 (voice)

Canada BBS's

Ontario:
PSI-Wordpro
Mississauga
sysop: Steve Punter
(416) 624-5431
hours of operation:
7PM-9AM weekdays
24 hours on weekend

Professional Software, Inc.
Mississauga
(416) 624-5431 (voice)

NORTEC
Toronto
asst sysop: Richard Bradley
(416) 782-7320
hours of operation:
7PM-9AM weekdays
24 hrs. on weekend

TPUG
Toronto
sysop: Tony Prijately
(416) 223-2325
hours of operation:
7PM-9AM weekdays
24 hrs. on weekend

Toronto PET User Group
c/o Electronics 2001, Ltd.
5529 Yonge Street
Willowdale
(416) 223-8400 (voice)

***NOTE:** A complete and up-to-date list on local bulletin boards across the country can be found in the Commodore Information Network (CIN) on CompuServe. This file is located in the C64 SIG in database #3 and is called PAMS. To get there, first log on to CompuServe and type GO CBM963 at any system prompt. Then type XA3 at the FUNCTION: prompt and then BRO PAMS.

broadcast television channels of videotext and graphics along side their regular signal. Interested? For more detailed information read Jim Gracely's article in this issue, page 47.

Micro to Mainframe

Micro-to-mainframe communication can be divided into two main uses. The micro can either serve as a remote terminal for a centralized mainframe or as an information receiver from a telecommunications network.

Remote terminals are used mostly by business. This type of terminal can be used for a broad range of applications such as inventory control, product tabulation and payroll.

A micro that is tied into the many and ever-growing number of information-providing services, such as CompuServe and Dow Jones, can bring a phenomenal amount of information into your home, business or school. Think of the host mainframe as a large database from which you can pick out the information you are interested in. To give you some indication of what's available, here is a list of some of the better known services, with enough information about each to get you started.

CompuServe (CIS)

This national telecommunications service offers information on a variety of topics ranging from news, weather and sports to electronic mail and shopping. CIS can be accessed 24 hours a day. During non-prime time and on weekends the rate is $6.00 per hour of connect time. During prime time the rate is $12.50 per hour. Any questions can be directed to CompuServe Customer Support at 800-848-8900.

Commodore Information Network (CIN)

A service for Commodore users located on the CompuServe network. CIN is a user friendly service composed of over 1000 menu-driven "pages" of text, a hotline dedicated to customer support and three national bulletin board SIGs (Special Interest Groups). For more information on CIN, see Tony Caramanico's article in this issue on page 38 or write to me at Commodore for more details.

The Source

A national information service much like CompuServe, offering information on a variety of topics. To contact The Source call 800-336-3300.

Comp-U-Store

A service of Com-U-Card International, Inc., Comp-U-Store is an electronic shopping mall that can be accessed either directly or through both CompuServe and The Source. Comp-U-Store members can access information on over 50,000 brand name products by manufacturer's name, model number or product features. They receive descriptive information, a price quote and the luxury of arranging delivery anywhere in the continental U.S. For more information call 800-843-7777.

Dow Jones News/Retrieval

A service that brings you up-to-the-minute news, information and stock quotations. In addition to news/retrieval, Dow Jones gives you access to the Wall Street Journal, Barron's, domestic and overseas news wires and much more. To contact Dow Jones call 800-257-5114.

GameMaster

This interactive network, located in the Chicago area, is dedicated to games, hobby discussions and information exchange. GameMaster is set up like a house. It is within this house that the gaming and information network is located. The house has been divided into many rooms, each having its own special feature, and is open 24 hours a day. There is a one-time entry fee and charges which are based on per-minute online time. If you have any questions, call the hotline at (312) 328-9009. The GameMaster also operates a 24-hour bulletin board at (312) 475-4884.

AGNET

A computer network that offers instant access to the latest agricultural news from the Economic Research Service (ERS). For more information contact AGNET at (402) 472-1892.

NewsNet

An online database service with over 175 authoritative newsletters plus UPI and PR news wire. Designed with the businessperson in mind, NewsNet delivers specialized information in the field of computers, taxation, accounting and more. Usage charges are $24 per hour for 300 baud, with a monthly minimum of $15. To get in touch with NewsNet, call 800-345-1301. C

Barbara Karpinski

Update:

Yes, another update but not just any update! The enormous growth of telecommunications has come upon us and will continue in decades ahead. And with this growth the Commodore Information Network (CIN) has been growing and changing to meet our users' needs.

This update explains the restructurization of CIN's three SIGs (Special Interest Groups). In the new structure two SIGs have been combined into one and a new SIG, The CBM Programming SIG, has been introduced.

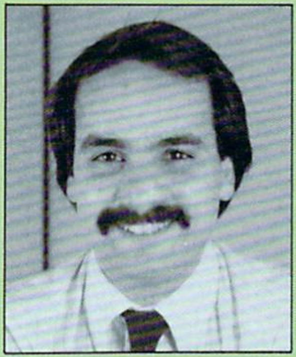
The new Programming SIG was developed to accommodate the growth and popularity of the Commodore 64 SIG, where so many members are interested in programming that we needed a separate area just for that topic. As part of the new SIG format, two databases from the Commodore 64 SIG—FORTH and CP/M™—have also been moved to the Programming SIG. The new Programming SIG is located at page number CBM-310, which used to be the location of the CBM Business Machine SIG. In addition, the CBM Business Machine SIG is now combined with the VIC 20 SIG and is located at CBM-962, which is the old VIC SIG location.

The following is a list of all the SIGs now on CIN, along with their database section names. Newly created SIGs are marked with an asterisk.

The Commodore Information Network

By Tony Caramanico Commodore Assistant Telecommunications Coordinator

	VIC 20—PET/CBM SIG	Commodore 64 SIG	CBM Programming SIG
0	Hotline (General)	Hotline (General)	*SIG Help Files
1	(not available)	Ask the Trade	*Children's Programming
2	VIC Recreational	64 Utility	*High Level Utility
3	VIC Utility	Beginner User	*Disk Programming
4	PET/CBM	*The 64th Dimension	FORTH Language
5	SuperPET	Manual Updates	*Patchwork Programs
6	CBM Mag. Archives	*The Arts	CP/M for the 64
7	CBM Mag. Programs	Games	COMAL/New Langs.
8	Telecommunications	CBM Public Domain	Telecommunications
9	CBM Nat'l Comp. Club	CBM Public Domain	(not available)



Tony Caramanico

The New SIGs

SIG Help Files has been set aside specifically to help all users access the information they want in the SIGs and aid them in conferencing, uploading, downloading, file submission and other areas that are related to the SIGs.

Children's Programming is devoted to programs for preschoolers and young children on all Commodore computers.

High Level Utility is dedicated to higher level utility programs for all Commodore computers (e.g., sprite editors for the Commodore 64).

Disk Programming contains numerous utility programs for the Commodore 1541 disk drive, such as disk backup programs and disk doctor programs.

The Arts contains music and graphics programs for the Commodore 64.

The 64th Dimension is an online newsletter where you can get up-to-the-minute news about the CIN and the SIGs. Specific nights are set aside for informal conferences among members with common interests. Current information on software and hardware for the Commodore product line is part of the database, along with a monthly Member Spotlight. The main objective is to help members interact with each other on an organized level.

Patchwork Programs contains patchwork text files and programs for patching programs.

To Find Us

I hope that all our CompuServe and Commodore Information Network members will join us in the new CBM Programming SIG and our other new data bases. The SIG page numbers are as follows:

VIC 20-PET/CBM SIG .... GO CBM-962
Commodore 64 SIG .... GO CBM-963
CBM Programming SIG .... GO CBM-310

If you're a Commodore modem owner and haven't accessed the Commodore Information Network lately, why not come and see all the changes we've made? There's bound to be something there for you. If you are not a Commodore modem owner, why not give telecommunications a try? There's a whole other world just waiting for you with hundreds of programs, utilities and information.

I hope to see you online. **C**

CP/M is a registered trademark of Digital Research

Illustrator—Verlin Miller

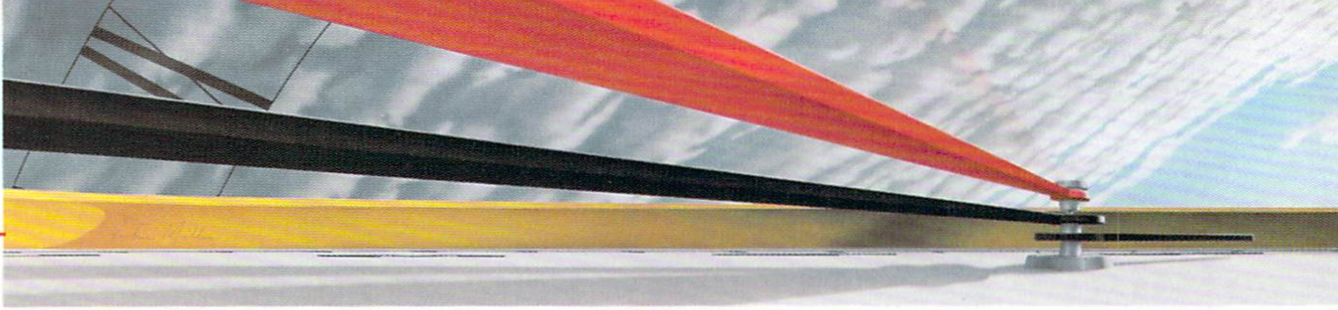
By deb! Christensen
Systems Operator, Commodore Information Network

When Time Counts

The Information Age is here. Well, at least that is what I read. The unfortunate part about the current status of this Information Age is that it can be very costly to use! No, the cost is not in acquiring the equipment or the software, but in the connect-time charges you must pay to access the major online services. If you use online services frequently, here's how to save money.

The hardware requirements for telecommunications are not exotic and Commodore prices make them easily affordable. For instance, if you set up a Commodore home computer, disk drive, dot matrix printer and modem, your total hardware investment is around $700. In addition, you will want a smart terminal program, too, but now that you can buy VIDTEX, CompuServe's executive terminal for the 64, it will cost you less than $40!! And I have not seen another terminal software package that is as versatile or as fast.

VIDTEX is a registered trademark of CompuServe, Inc.



As you can see, for well under $800 you can have a complete Commodore system for accessing any online service. This is an astonishingly low cost, and you can feel very lucky to have paid so little for so much.

Using this inexpensive system, you can bank, mail, play games, take classes, attend hobby and professional conferences, research, do your shopping, get the latest stock quotes, find instant news and even work from your home. It sounds wonderful! But, then you get the bad news—you have to pay for each minute you are connected. And some of the national networks, such as CompuServe and Dow Jones, can cost anywhere from $6.00 per hour to $72 per hour.

So, whether you are using a local videotext, a home-run bulletin board or a national service, there seems to be a critical need to conserve time. But in an area where time is so crucial, telecommunications has its own kind of time reference. Minutes can pass by on your watch, but it will seem like no time at all as you watch your video monitor. I have jokingly referred to this as a "time warp", which is not all that far off the base. To us "online junkies", the clocks in the real world seem to have little bearing on the time that passes through the modem! Unfortunately, the people who send out the phone bills and the connect charges have a very firm hold on real time.

You are the only one who can decide how much time is worth what cost, but since saving time —and thus phone and connect-time costs—is so crucial when you are online, let me show you how you can make a little better use of that time.

EasyScript/VIDTEX

First, use a smart terminal package. A good one, like VIDTEX, can both save you time and let you make better use of the time (and money) you have budgeted for the Information Age. Next, armed with this smart terminal program and a good word processor, you can accomplish many editing and composing tasks offline. For instance, this article was written using the Commodore *Easy Script* word processor offline, then uploaded to Commodore using VIDTEX. Both of these software packages are well written, well documented and completely compatible. (They are also two of the best buys on the software market right now.) Let's take a closer look at how you can use them to reduce your online time.

Just so that we will know what we are talking about, here are definitions for a few of the unique telecommunications terms I am using.

Downloading
The process of receiving incoming information into your computer.

Uploading
The process of sending information to another computer. This can be text or programs.

Host
The computer that you call, i.e., CompuServe is the HOST system.

ASCII
The standard eight-bit values for the characters displayed on your screen.

PET ASCII
The Commodore values for the characters (they are not standard ASCII).

RAM Buffer
The space set aside in your terminal program to use to store the incoming data.

Meta
The special key in VIDTEX which controls all the local functions of the terminal program, such as opening buffers, defining controls, etc.

Protocol
The special sequences which are used to check for errors in the up and downloading processes. CompuServe uses "B protocols" with VIDTEX to insure accurate delivery of files. These protocol sequences are initiated by the host and VIDTEX—you do not have to do anything.

Are you ready to log in? I sure hope so, because here we go!

```
**Data Carrier Present**
C

User ID: 70007,577
Password!
CompuServe Information
Service
11:27 PST Monday    09-Apr-84 P

You have EMAIL waiting.
OK
r email

CompuServe          Page EMA-1

Electronic Mail Main Menu

1 Read Mail
2 Compose and send mail

Last menu page. Key digit
or M for previous menu.
! 1

CompuServe          Page EMA-3
1 Buzz/CRAZY PRINT
  STATEMENTS
2 DICK MARKS/COURSE
  ON BASIC
3 Brooklyn-mike/higher
  education
4 jennie/computer class!
5 JMcA/REL-COPY prg.
6 JMcA/..continued REL prog.
  Last menu page. Key digit or
  M for previous menu.
```

Now, this is where I can read my mail. I open my buffer and read the one I want to see. (You don't really want to read my mail do you?) After I have read it, I close my buffer and store it on my disk. But I don't waste valuable time now by trying to answer it while I am connected to CompuServe. Log off and collect your thoughts, boot up your *Easy Script* and compose your reply at your leisure. By the way, there is an added benefit, even aside from saving time, to composing text using *Easy Script,* rather than relying on the online text editor. You get full-cursor editing capabilities using *Easy Script,* whereas online text editors are usually line-oriented and leave a lot to be desired.

Before we go any further, however, let me add a word or two about files! SEQuential files are one of your best friends! Standard word processing files are stored in PET ASCII on your disk, with returns only where you put them. To be able to use *Easy Script* effectively with any electronic service, you need to be sure that there are carriage returns inserted after every line. The lines are typically either 40 or 80 columns. You can accomplish this very simply. Just be sure to hit RETURN at the end of the desired number of characters! Your word processor usually will insert these RETURNs for you when it is printing out to the printer, because it is a function of the software to output a formatted hard copy. Leaving all the text that is stored in memory and on the disk in paragraph form saves space.

Remember, EMAIL also needs carriage returns every so often, so be sure that your composed message has them on each line. I use 70 or 80 columns for EMAIL, then press RETURN.

After your text file has been saved, you just boot up VIDTEX and log back onto CompuServe. You have two options here. The first is the easiest. There is a special File Transfer program that resides on CompuServe and can be accessed from your Personal File Space. It is called FILTRN and is a protocol transfer for either uploading or downloading, for VIDTEX users only. At the OK prompt or the PCS71 menu, request a file transfer. In the command mode, this is: R FILTRN.

You will be asked whether you want to upload something or download something. Obviously, we want to upload now, right? After you answer the questions, and provide a file name, your file on the disk will be transferred directly from the disk to a file on CompuServe. When that is done, you have a file which can then be transferred to EMAIL from the EMAIL menu by choosing the option number three, "Create a File from your Disk Space".

Now, that was pretty painless, right? Just remember that whenever possible, compose your replies offline!

Buffer Upload

Another quick way to send EMAIL is to use a buffer upload rather than a file transfer. Follow the same guidelines for composing the letter as outlined above. Hit RUN/STOP and RESTORE to see the title page again and set up with printer option number one. This will allow you to output an ASCII file to your disk! Now, instead of saving the file within a standard *Easy Script* file, use the following sequence of strokes: <F1> <0> <s>.

Files saved in this manner are placed on your disk just as they are formatted on the printer. That is why choosing printer option number one allows you to write ASCII files. And if your message is longer than one page, it will even provide the right number of carriage returns for the page breaks! To output a file like this in PET ASCII, you just use the "0 (CBM)" printer option. However, *Easy Script* will insert a couple of weird

characters at the beginning of each line—a "q0"—so you will have to load the new file in and search-and-replace them out of the file. The ASCII option (number one) does not give you anything but your text.

The proper use of ASCII and PET ASCII files is critical when you are using VIDTEX, but fortunately it's very simple to figure out. A protocol transfer like FILTRN or the SIG (Special Interest Group) command, (UPL), expects to find a PET ASCII file. But, if you have saved your file as an ASCII file, VIDTEX translates it anyway, and you are left with garbage in your file.

On the other hand, if you are going to use a buffer upload and transmit each line from your RAM buffer, then VIDTEX doesn't really know how to read a PET ASCII file, and needs that file to have been written out in real ASCII so that the upper and lower case letters are all fine. Buffer uploads are very nice for other services and for leaving messages on the CBM SIGs.

SIG Messages

Speaking of that SIG, let's see if I have any messages waiting there!

```
G PCS156

CompuServe        Page PCS-156
Request Recorded,
One Moment, Please
Thank you for Waiting

Welcome to Commodore 64
SIG, V. 2C(10)

Name:        deb! 70007,577
Last on:     09-Apr-84  10:57:31
High msg#:  57342

You are user number 250551
```

```
System contains messages
57571 to 57958

You have a message waiting:

#: 57958 (P) Sec. 4—The 64th
Dimension
Sb: broken thread
     09-Apr-84 11:24:27
Fm: SYSOP/Howard Rotenberg   7007,575

This message has been
marked for retrieval with
the RM Command

Function:rm
```

Well look at that, I DO have a message waiting. (No, you don't get to read this one, either!) If the message requires a good size reply, then I do exactly what I did with the EMAIL. I save it, print it out and compose my reply with *Easy Script.* SIG messages, however, do not have a FILTRN, so you've got to use a buffer upload. Here are a few more hints on creating replies to SIG messages:

1

Don't use any blank lines! These will cause the SIG line editor to think you want to exit.

2

The SIG editor tries to outsmart you all the time. It will format your message to conform to the defaults of the person who is reading it. For instance, if you want to start a new line, you must use either a period or a space as the first character after the line number prompt. Just because you press RETURN and get a new line does not mean that the SIG will start a new line for you when the person reads it. So, remember this: to start a new line use a period or a space as the leading character. The period will not show up in the finished message; it will just force the SIG editor to do things your way!

3

If you want your message to be in sequence with the topic of the one you are REplying to, use the RE command at the Function Menu. Just be sure you have the message number handy so you can enter it after you type in RE. This is a very handy time-saving technique. It also preserves the thread of the conversation to make it easier for other people who are using the scanning and searching commands to follow what's going on.

4

When you are ready to send a message, clear your RAM buffer and use a META L to load the ASCII file into it.

5

After that, use META Y to transmit the text. For SIG messages, use a colon for the prompt when VIDTEX asks you for it. Then your reply will be transmitted line by line, since VIDTEX sees the colon that CompuServe sends at the beginning of each new line.

Making the very most of your online time should be your first consideration when you're using any telecommunications service. After all, I would much rather spend time enjoying the Information Age than having to fight with online text editors and

higher bills!

There are even some programs that can help you do much of your offline editing if you don't have a word processor that stores standard SEQuential file. For instance, using VIDTEX you can download the following programs, ready to run, from the Commodore 64 SIG on the Commodore Information Network, and use them to help you while you are offline:

TINYWO.IMG Tiny Word, a small text editor in BASIC.
VIDCON.IMG Will translate PET ASCII files to ASCII.
TEXTED.IMG Another text editor, BASIC.
TXTED-.IMG The compiled version of TEXTED.IMG. Much faster but requires another file to run.
SPLIT.IMG Will split up large files so that you can edit them on your word processor.
SEQPRN.IMG Will print out a SEQ file to your printer, filtering out control codes.

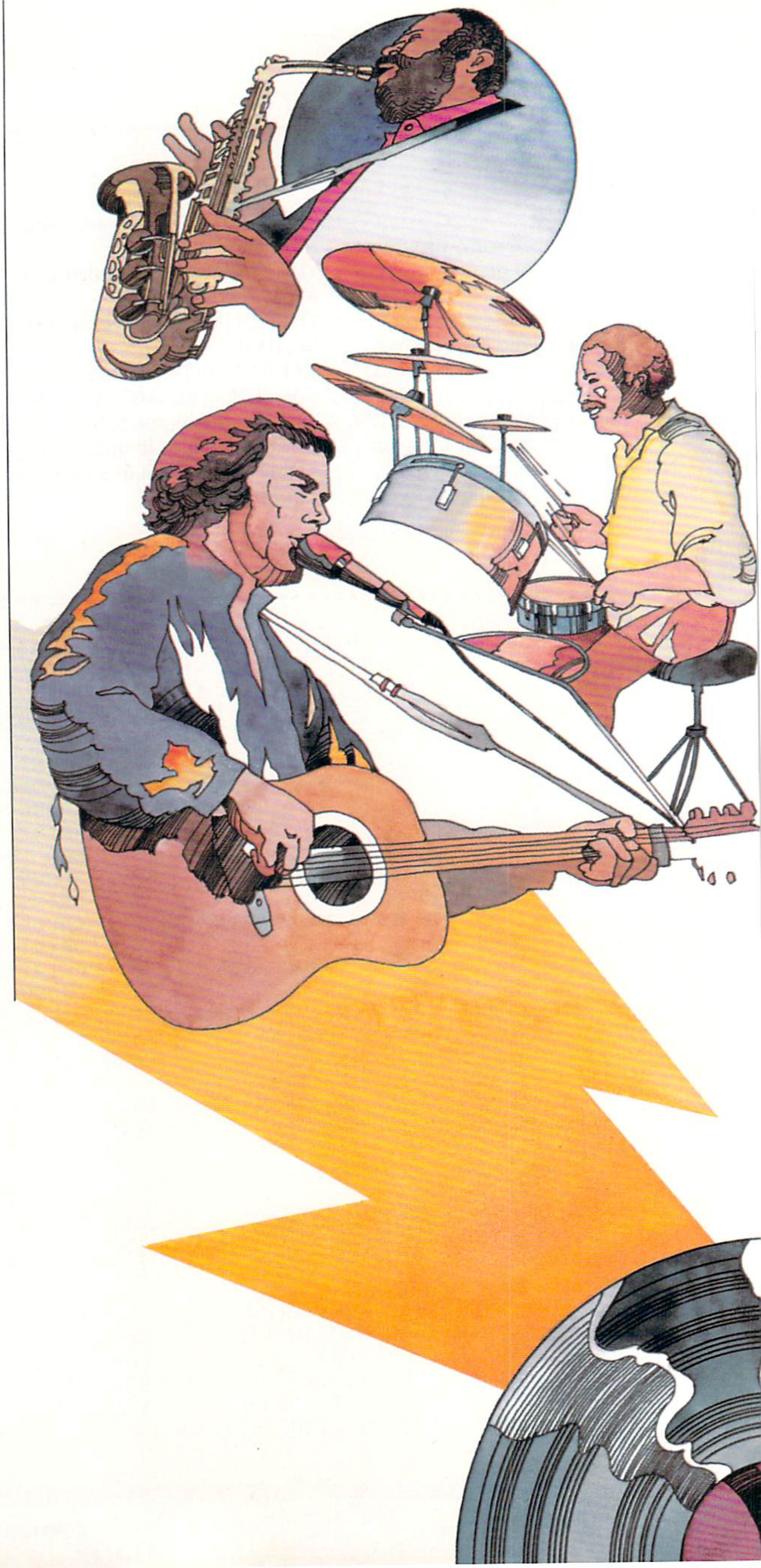
Come and visit with me some time on a Commodore 64 SIG conference! It will be nice to meet you!

And for those of you who need it, here follows a small program which will READ any SEQuential disk file for you. C

```
1 REM ******************************
2 REM *                            *
3 REM *        READ ANY FILE       *
4 REM *                            *
5 REM *            BY              *
6 REM *                            *
7 REM *           DEB!             *
8 REM *                            *
9 REM ******************************
10 GOSUB 500:GOTO 18
18 PRINT CHR$(147)
19 GET#8,A$
20 GET X$
21 IF X$<>""THEN GOSUB 50
22 IF O$="P"THEN IF ST=0 THEN PRINT#4,
   A$;: GOTO 19
23 IF ST=0 THEN PRINT A$;:GOTO 19
25 IF ST<>0 THEN 200
50 IF X$=CHR$(19)THEN GOSUB 70:RETURN
   :REM PAUSE PLEASE AND LET ME CATCH
   UP!
52 IF X$=CHR$(16)THEN GOTO 200
   :REM DON'T WANT TO READ THIS
   ANYWAY SWEETIE!
53 RETURN
70 GET X$:IF X$=""THEN 70
75 RETURN
200 PRINT A$:IF PEEK(152)>2 THEN
    PRINT#4,A$
205 INPUT#15,N,N$
206 IF N>1 THEN PRINT
    CHR$(17)CHR$(17) "[SHFT D,SHFT I,
    SHFT S,SHFT K] [SHFT E,SHFT R2,
    SHFT O,SHFT R]!! #"N;N$
210 CLOSE 4:CLOSE 8:CLOSE 15
    :PRINT"[SHFT A]LL FILES CLOSED"
220 PRINT"[SHFT A]NOTHER [SHFT F]ILE?
    [SPACE2](Y/N)"
221 GET A$:IF A$=""THEN 221
222 IF A$="Y"THEN 250
223 IF A$<>"N"THEN 221
225 END
250 GOSUB 540:GOTO 18
499 END
500 PRINT CHR$(147);"[7 DOWN]"
    : POKE 53280,6:POKE 53281,15
    :POKE 646,6
510 PRINT CHR$(14)"[SPACE13,SHFT R]
    EAD [SHFT A]NY [SHFT F]ILE"
512 GOSUB 520
515 PRINT CHR$(17)CHR$(17)CHR$(17)
    CHR$(17)CHR$(17)"[SPACE10,SHFT C]
    OPYRIGHT 1983 BY"
516 PRINT"[SPACE12]DEB CHRISTENSEN"
    :GOSUB 520
519 GOSUB 520:GOSUB 540:RETURN
520 FOR I=1 TO 1200:NEXT I:RETURN
540 INPUT"[SHFT N]AME OF INPUT FILE";
    F$
541 F$=F$+",S,R"
545 CLOSE 15:OPEN 15,8,15
    :OPEN 8,8,8,F$
550 PRINT:PRINT: PRINT"CONTROL <S> TO
    PAUSE WHILE READING TEXT!"
551 PRINT"CONTROL <Q> TO CONTINUE."
552 PRINT"CONTROL <P> TO SKIP."
554 PRINT:PRINT: PRINT"[SHFT O]
    UTPUT TO:[SPACE4][[SHFT P]]
    RINTER OR [[SHFT S]]CREEN?"
555 GET O$:IF O$=""THEN 555
560 IF O$="S"THEN RETURN
570 IF O$<>"P"THEN 555
575 CLOSE 4:OPEN 4,4,7:RETURN
```

Radio Tele-Text With The GRAIL

By Jim Gracely
Technical Editor



Illustrator—Judy Newhouse

In the state of Vermont, in the town of Montpelier, a radio station called WNCS, 96.7 FM, broadcasts contemporary adult rock music 24 hours a day. But hidden inside that signal is a most unusual subscription television channel, providing maps, entertainment, news and weather reports—all generated on a Commodore 64.

A small (for now) company in Vermont called the GRAIL Corporation is breaking ground in a new field of telecommunications, more correctly referred to as radio telecommunications. This company is using a combination of Commodore 64s, modems and WNCS FM to create an information network in the Mad River Valley of Vermont.

The name of the company, GRAIL, is an acronym for Greater Resort Area Information Link. The resort areas covered by the system presently include Stowe, Sugarbush and the surrounding towns. The basic concept behind the service is to provide information on weather conditions, things to do and places to visit within the viewing and listening area of the person using the system.

A television in any room of a hotel or condominium that has subscribed to the service can receive the GRAIL on channel 4.

When that channel is tuned in, the normal WNCS radio programs are received as well as the maps, text and graphics of the GRAIL transmission. For the person staying in one of these rooms, the GRAIL acts as a radio station, a 24-hour weather and skiing condition station and guide to entertainment and attractions in the area.

The Service

The service provided by the GRAIL systems is a one-way communication information link, supplying a variety of information about the local areas. The information base is organized into 40 screens or pages. Each screen is displayed for 25 seconds, which means that all information is displayed every 16 minutes.

John Eddy, President of the GRAIL Corporation, likes to refer to the service as "viewer friendly". He feels that many of the information services in cities across North America are too gaudy or just plain irritating to watch. He wanted to create screens of information that were easy on the eyes and yet could catch your eye if desired.

John's solution to these problems was twofold. First, for any screen that displays strictly "printed" information or text, the background is white and the letters are black. John felt this was easiest on the eyes. A second advantage to this color scheme is that it is easy to make anything important stand out. As John pointed out, if there were a bad ice storm coming he could have that message

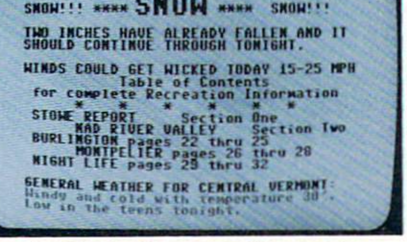


Photo by Nancy Wong

The GRAIL gives Vermont visitors up-to-date weather information.

For the person staying in one of these rooms, the GRAIL acts as a radio station, a 24-hour weather and skiing condition station and guide to entertainment and attractions in the area.

printed in red rather than black, and viewers' eyes would pick it up quickly. This ability for emphasis would be lost on a screen that was always five or six rainbow colors.

The second solution, for non-text screens, involves the use of the 64's graphic capabilities. John has included in this service a number of area maps, all designed on the 64 with full color graphics. In addition, color graphics and drawings are used to emphasize special events and to enhance advertisements.

One of the other features of this service that makes it "viewer friendly" is the fact that it can be updated almost instantly. According to John, the company was updating the system twice a day this past winter—once in the early morning and once late in the afternoon. He imagines that this will decrease to once a day in the summer, under normal circumstances. But if something warrants an additional update, it is easily implemented.

The service, as it is currently set up, contains 40 screens of information. Of this number, 22 are devoted to resort area information and entertainment. The remaining 18 pages are advertising. The policy of the GRAIL is to accept no advertising for hotels, condos or other lodging. As John explains, everyone watching the service is already staying at some lodging facility, and allowing competitive advertisers would create conflict within the system.

The information is divided into seven sections, each occupying between four and eight pages. The introduction uses four pages, with a full "front page" and a summary of important news and events. Again, the ability to update the service is a big advantage here. On inclement days, for instance, attention on the system can be focused on alternate recreational activities.

In addition, there are area reports on Mad River Valley, Stowe, Montpelier and Burlington. These area reports contain information, maps and advertising related to the individual area. In the winter time, this section places the most emphasis on skiing and other winter sports. In the summer this section will include information on sailing, fishing, horseback riding and everything else that the areas have to offer.

Two sections of the system are called "Night Life" and "Vermont—The Extra-Ordinary State". Night Life contains information on movies, bands, concerts, plays and any other special events. The Vermont section contains information on covered bridges, waterfalls, barns and other places of interest.

How It All Began

The idea behind the GRAIL was born during the Winter of 1981. John Eddy, who also owns Mad River Video, explained the conception of the GRAIL system to me in a telephone conversation. Part of John's work with Mad River Video included delivering prerecorded movies to various homes and condominiums. Whenever he delivered a movie to a room occupied by visitors, he was inevitably asked "What's the weather supposed to be like tomorrow?" and "Have you heard how the skiing conditions are today?".

Being an entrepreneur, John immediately recognized the potential for a subscription information service that would answer these questions for people. Look-

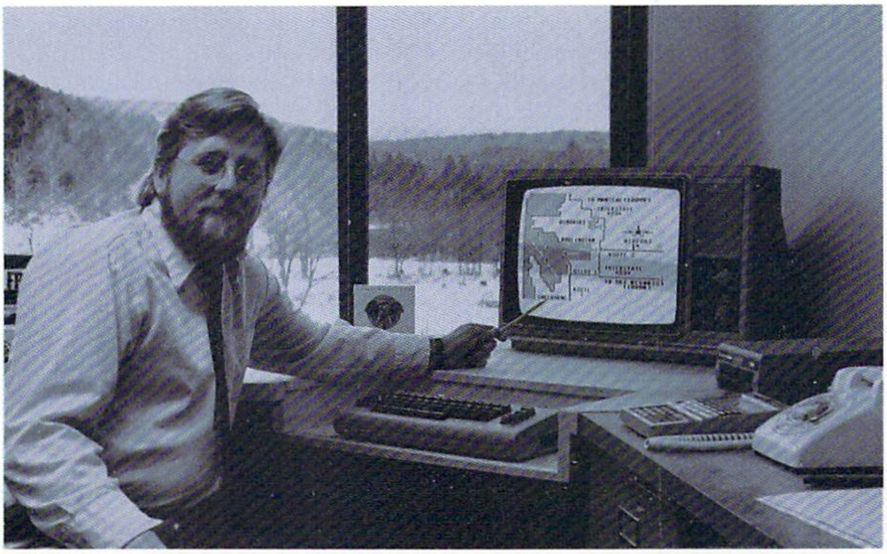


Photo courtesy The Valley Reporter

John Eddy, President of the GRAIL Corporation, broadcasts maps of the Vermont resort areas as one of his company's services.

ing around the rooms he visited, he tried to envision a way to get that information into those rooms. After ruling out the telephone, the front door (mail or carrier delivery) and the chimney, he noticed the television. The idea of using the television seemed the most promising way to supply information to the rooms, but it also presented some difficult problems.

During the summer of 1982, after mulling over the basic scheme, John teamed up with Howard Ginsberg, a broadcast engineer and president of Communications Engineering, Inc., in Essex Junction, Vermont. As they discussed the various problems, two seemingly incompatible solutions emerged. The first was the idea of using a personal computer to design and digitalize the information, and the second was the concept of broadcasting that information through an FM radio station.

The technology was already available to broadcast a subscription audio signal through an FM station. This is called SCA and is used by a number of "store music" companies. The combination of this technology and personal computers, however, had not yet been tried out—especially the transmission of a video rather than audio signal.

The search was on for a computer and system software that could handle the test and graphic screens required and was still priced low enough for the application. John says that during this search he was actually thrown out of two computer stores amid cries of, "It can't be done!" After considering a number of computers, they finally made the decision to use the Commodore 64 throughout the system. John says he had talked to engineers and computer hackers and everyone seemed to agree that the Commodore 64 offered the best graphics, memory capability and price for this application. Now it was time for the program.

After confirming that there was no commercially available software for this application, John enlisted the services of fourteen year-old Wilson Snyder of South Burlington, Vermont. Starting in the fall of 1983 and working diligently on the program for almost three months, Wilson created the complete software package. It allowed formatting of screens and the contents contained on them, included a telecommunications package to convert the screens into streams of data which could be sent through a modem, and was run entirely on a timing schedule. After working with this software for many months now, John has begun referring to Wilson as the "Chopin of Computers".

The next step in the process was to find an FM radio station that would lease the SCA frequency to the GRAIL. It was also important for this radio station to be centrally located between the large resort areas and have a powerful enough signal to reach them. With these criteria and a list of stations in the central Vermont area, the selection of the radio station was simply a matter of deduction. The station that was chosen (and accepted) was WNCS, 96.7 FM, in Montpelier.

The final (and continuing) step was to find resorts, hotels and condominiums that recognized the benefits of this system and would subscribe. This turned out to be one of the easier aspects of the project. In fact, before the system was even powered up, over 500 rooms in central Vermont were signed up for the service.

Then on Friday, December 16, 1983, two years after the initial conception, the Commodore 64 at the GRAIL was turned on, the telephone line to WNCS was connected, the SCA at the station started transmitting and the service came alive.

System Description

The GRAIL system uses a combination of telecommunication and radio communications to accomplish its ends. This is a unique application of a Commodore 64, used in conjunction with standard radio broadcasting techniques. The modem provides the necessary interface between the digital and analog worlds. Here is a brief description of the entire system and its various components.

The entire information content of the GRAIL originates within a Commodore 64, located in Warren, Vermont. This computer is

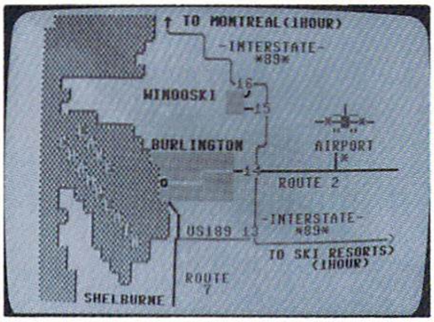


Photo by Nancy Wong

The system's maps help visitors plan their day.

running the GRAIL system software that controls the information being displayed at any given moment, along with the timing between individual screens. An additional function of this program is to handle the telecommunications routines that actually transmit the data.

Each screen of data is sent through a modem and telephone wires to the FM radio station WNCS in Montpelier. Depending on which particular screen of data is active, the system can be sending anything from simple text to full color screen graphics. This capability was made possible as part of the custom programming created by Wilson Snyder.

A device called an SCA encoder/transmitter at the radio station receives the data from the telephone line. This piece of equipment is part of what makes this system so unique. An SCA encoder takes an audio input, such as that from a telephone line, and changes it into a radio (SCA) signal of a very specific frequency (67 KHz). The transmitter part of the SCA simply relays the radio signal to the main FM transmitter. The FM transmitter multiplexes the SCA signal with the FM radio signal and the resultant signal is then broadcast throughout the normal FM reception area. For WNCS this includes both the Stowe and Sugarbush resorts as well as the cities of Montpelier and Burlington.

If a resort or condominium has subscribed to the GRAIL, they will be equipped with an SCA receiver. The SCA receiver strips the SCA signal off the radio broadcast and sends it to a receiver Commodore 64. A special program within this 64 translates the data from the SCA back into screens of text or graphics. The output of the receiver 64 is connected to all the televisions in the rooms of the resort or condominium. A television in any room which is turned onto channel 4 will see the GRAIL screens displayed, with full graphics and color.

An additional benefit of using Commodore 64s in this system is in the ability to use the normal radio broadcast, as well as the GRAIL screens. The regular radio signal is combined with the special video signal, so the

Glossary

FCC

Federal Communications Commission. This is the branch of the U.S. government that regulates all forms of communications. Everything from spaceman walkie-talkies to cable satellite transmissions are under the auspices of the FCC. Even computers must be approved by the FCC before being marketed. The regulations for computers center around the connection with a television or monitor and the interference (or electromagnetic radiation) that the computer may produce.

Frequency

The number of complete cycles of a sound or electrical waves per second. Normally the word frequency is used with the 64 in association with music. This is the audio portion of the spectrum. When the vibrations are very fast, they are inaudible to the human ear and are used for such things as AM and FM radio broadcasting.

GRAIL

Greater Resort Area Information Link. The acronym used by the GRAIL Corporation for their radio telecommunications system.

Hertz

A unit of frequency equal to one cycle per second and abbreviated Hz. A number preceding the word refers to the number of cycles per second. 60 Hz represents 60 cycles per second. An "M" preceding the word represents one million Hz or one MegaHertz. A "K" preceding it means one thousand Hz.

Information Link

Similar to a network, although the purpose of the communication is strictly informational. It also usually applies to a one-way system in which the user has no control over what is displayed.

Network

A number of people sharing a common information base or communications system. The idea of a television network is the most common application of the word.

SCA

Subsidiary Communications Authorization. This is the FCC regulation allowing an FM station to broadcast both its normal signal and a special background music or subscription program signal. The most commonly encountered use of SCA is for "store music".

Subcarrier

This is the technical term for a special signal (such as SCA) being broadcast at the same time as a normal signal. The word subcarrier is used because the special signal is "under", or less powerful, than the main carrier. C

radio station broadcast can also be received through the television at the same time. In fact, John claims that the quality of the radio station signal is much better than he had anticipated.

This is a simplified version of the way the GRAIL works. In the interest of both space and the security of the GRAIL system, this is the most that I can explain.

A Look Into the Future

What does John Eddy see as the future of the GRAIL or the technology that he has introduced? John shared some of his thoughts, hopes and predictions with me.

The aim of the GRAIL Corporation is to sell the complete GRAIL system to other resorts and cities as a viable tourist and visitor's aid. According to John, everyone benefits because of the way this system is set up. The resort or city using the system has the ability to provide special attention to its places of interest and can promote the area as a good place to visit or spend a vacation. The individual hotels or condominiums offering the service to their guests have a unique addition to the standard room radios and visitor guides. The FM station that broadcasts the signal benefits from both a licensing revenue on the SCA frequency and increased listenership, resulting in increased advertising revenues.

John also believes that now that the groundwork is broken in this field, other areas of broadcasting will probably pick up on it. Both cable and LPTV (low power television) stations can use the computer technology to create both screen graphics and text screens, reducing the cost of these features tremendously.

A Last Few Words

The use of Commodore 64s in this application is only one of the many possible ways that personal computers can be used on the leading edge of technology, in fields as diverse as radio, music, dance and weather predicting. It should be an encouragement to computer "watchers" to see that a small group of people can still bring innovation to the mass-market world of computers. It is somehow satisfying to know that the application of personal computers is not limited just to updating old technologies, but is just as well suited for the creation of new ones. **C**

Jim Gracely

What is an SCA?

SCA stands for Subsidiary Communications Authority and is the FCC regulation that allows an FM radio station to broadcast a special subcarrier signal along with its normal signal.

This authorization used to be obtainable only through a special license granted by the FCC. In the beginning of 1983 the SCA was deregulated, so that special licenses were no longer needed for an FM station to use an SCA signal. This deregulation has effects throughout the United States in the use of FM stations for various subscription services. For instance, some FM radio stations have started selling services such as radio paging and medica data transmission.

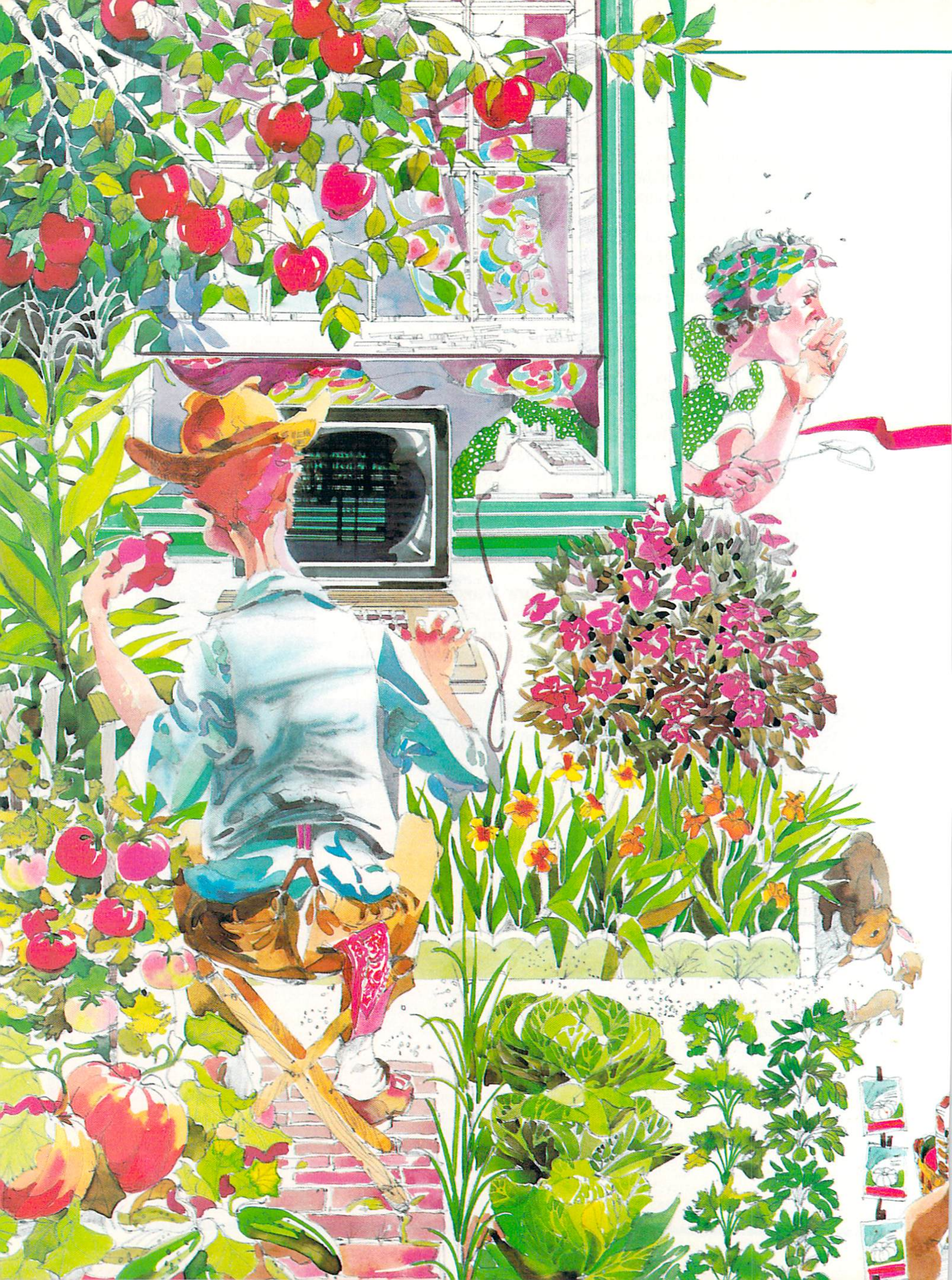
The basic concept behind the SCA is that a main or carrier signal can have one or more subcarrier signals, which are located at a frequency just a little above that of the carrier. For example, the GRAIL is using an SCA signal at 67KHz. The carrier signal at WNCS is 96.7MHz. This means that the SCA signal will be located 67KHz higher than the 96.7MHz signal, or at 96.767MHz.

An ordinary FM receiver cannot decode the left and right channels of music from a stereo FM station, so the result is that the receiver just mushes them together. In the same way, a stereo FM receiver cannot decode the SCA signal from the left and right channels of an FM station broadcasting an SCA signal.

The SCA decoder is like a super tuned receiver. It can pick out the SCA signal at 96.767MHz and demodulate it into an audio signal. The SCA receiver or decoder is not an overly expensive piece of equipment. Some companies even sell SCA boards that add SCA capabilities to an FM stereo receiver. One of the other applications of the SCA currently being tried out is a radio paging system. In this case the SCA receiver is about the size of a pocket radio. The doctor or other professional carries the receiver and hears a beep or some other noise when the page is activated.

I would like to thank Howard Ginsberg of Communications Broadcasting, Inc. of Essex Junction, Vermont, and Stan Kosciuk of GTE in New York for their help in explaining SCA. For further information on the subject, refer to the telecommunications books in your local library. There are two periodicals focused on broadcasting that may be of interest to those wishing to watch this new technology. They are: *Radio News* and *Broadcast Marketing & Technology News*. **C**

Jim Gracely



Commodore Communications

From a writer's point of view, telecommunications is the greatest thing since the invention of the typewriter. Here's a description of how a modem works and what you can do to make telecommunications work for *you*.

By Mark Brownstein

The phone rings in Los Angeles. *Nimrod Journal*'s editor is frantic —the article on painting with old food hasn't arrived and he needs something written right away. Can I come up with a short piece on, say, decorating your house with garbage? And can he have it in his hands, in New York, in two hours?

Of course he can. I take a quick look around my office for inspiration, and start banging on the keys. About an hour and a half later I'm done with another masterful piece of schlock and, inside of the two hour limit, my article goes from a spot on the inside of the editor's head to hard copy in his hands—from Los Angeles to New York—and, again, all in the space of less than two hours.

While I was working on the piece, I had to find some of the ornamental uses of brussels sprouts, so I looked through some of the reference materials on file at a library in Montana. And, of course, my printer had no ribbon, so nothing could be printed out.

Is this scenario impossible? Did it really happen? No, the events were changed to satisfy my weird sense of the ridiculous. But the only absurdities were the *events* and not the situations or the devices used.

Illustrator—Robert Neumann

In the course of my writing, I've literally made immediate transfers of text from my computer to a computer in New Hampshire, all in the space of very few minutes. An investor in San Diego, California, can get current stock reports from the major New York stock exchanges. A housewife in New Orleans, Louisiana, can order a can opener by computer from a store thousands of miles away (of course, it will be shipped by normal routes, but the order is handled immediately).

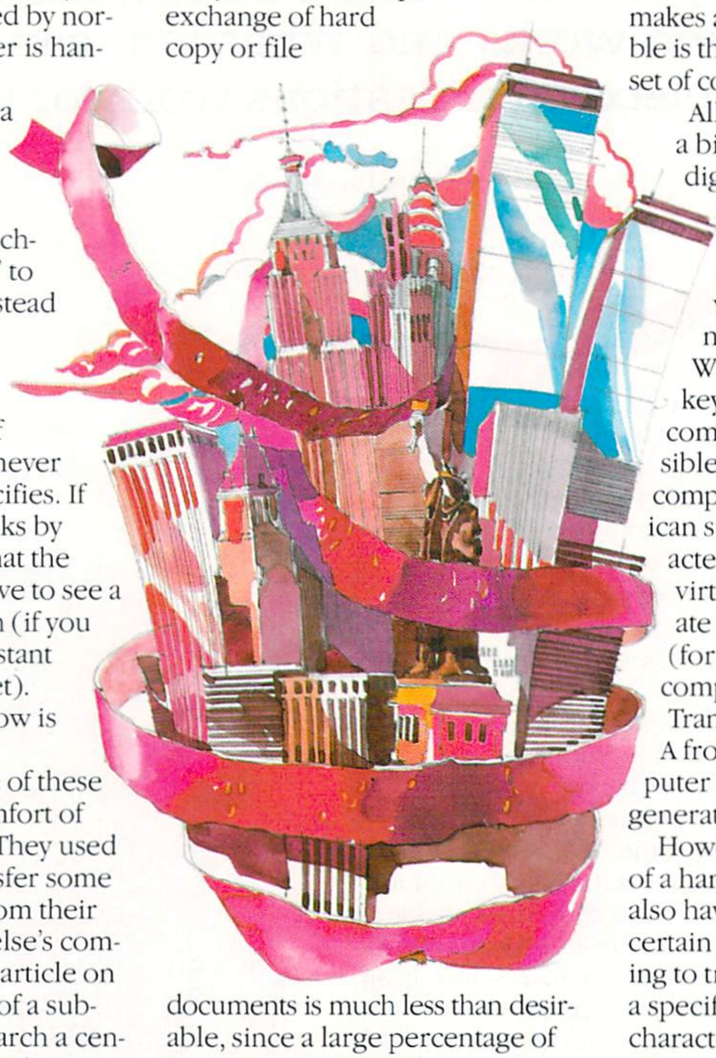
At homes in California and New York, and probably other states by the time you read this, people will be switching from "bank by mail" to "bank by computer." Instead of writing checks, their computer will direct the automatic issuing of checks (or transfer of funds) payable to whomever the account holder specifies. If your employer also banks by mail, it is conceivable that the only time you'll ever have to see a bank is to get some cash (if you don't want to use the instant teller at the supermarket).

What's happening? How is all this done?

It's obvious that none of these people ever left the comfort of their homes or offices. They used their computers to transfer some forms of information from their computer to someone else's computer. In the case of my article on decorating, I made use of a subscription database to search a central computer's memory about my subject of interest; I wrote the article on my computer, and then used my computer to transmit that article to an intermediate data service which, in turn, retransmitted it to the overly anxious editor on the other coast.

The other characters in the above scenarios similarly made use of their computers in connecting with larger (and maybe even smaller) computers for transfer of information. The overall process of connecting computers together is one given the general term *telecommunications*.

Telecommunications is one of the marvels of modern computing, allowing users virtually anywhere to bridge limitations of time, space and, to a large part, incompatibility between computers. The exchange of hard copy or file



documents is much less than desirable, since a large percentage of computers in use today, particularly the low-end machines (and primarily excluding PC and PC compatibles) have incompatible disk formats (an Atari disk won't work on a Commodore drive, or vice versa, and neither will work on a Texas Instruments computer, etc., etc.). Mailing incompatible disks is slow and there is a risk of accidental loss or damage to a disk. Mailing hard copy printout will require re-entry for any changes and is thus too time consuming.

Telecommunications overcomes most of these incompatibility problems among computers —an Atari can talk to a Commodore; a Commodore can talk to a multi-million dollar mainframe computer; and that high-priced monster can then talk to a lowly Timex Sinclair computer. What makes all these machines compatible is their use of an agreed upon set of communications standards.

All computers operate using a binary system of bits (binary digits) that are, literally, on and off switches built into the computer. Eight of these bits constitute a byte, which contains 2^8 possible numeric combinations.

When you hit a key on a keyboard, it causes a specific combination of those 256 possible choices to be stored in the computer, according to an American standard for all key characters, ASCII. It is the fact that virtually all computers generate the same ASCII characters (for text, at least) that makes computers so compatible.

Transferring the code for letter A from a Commodore computer to an Atari computer will generate the letter A on the Atari.

However, Commodore is one of a handful of manufacturers who also have unique ASCII codes for certain graphic symbols. Attempting to transfer the ASCII code for a specific Commodore graphic character will result in that character on another Commodore, but something less than predictable on other machines.

Now that we see that computers use standard codes for most alphanumeric characters and we understand that transferring code from one computer to another would generate an appropriate matching symbol, this still doesn't explain how the transfer is made. You can't

just hook one wire in your computer to another wire in the other computer and expect them to communicate.

To accomplish communication between computers, they must first speak the same language. Although ASCII, technically, forms the words, the ASCII codes in one computer must be converted for transmission to the other computer and, once these codes reach the other computer, they must be converted back to an understandable form of ASCII.

The internal pattern of electrical pulses that go on inside a computer can't be sent directly out to another computer—they wouldn't make sense. Therefore, a device must be used to *modulate* the signal so that it can be transmitted through phone lines or through cables. At the other end, another device must *demodulate* these signals, translating them back into a form that the receiving computer can understand. Since data transfer goes in both directions, such a translating device must both *MO*dulate and *DEM*odulate the signals (giving rise to the name of the device which does just that—the *MODEM*).

But perhaps I've oversimplified. If you were attempting to connect two computers that were two feet apart, you may be successful in connecting certain parts of both computers together to allow them to communicate. However, if you wanted to send the information through a phone line, without modification, you'd fail miserably for a number of reasons. First, the computer's output would not be strong enough to put through the phone lines. Second, the signal would not be understandable to the phone system and couldn't be accurately reproduced at the other end.

When AT&T first began to develop communications over wires (remember Western Union?), they had to devise a system that worked. The current, most-used adaptation of that standard (especially for home computers) is called RS-232C. RS-232C is a standard method of converting computer-generated ASCII code to a form that can be transmitted and accurately received and reassembled. On Commodore computers, this is the primary (if not the only) interface used for communications.

Transmission Standards

The RS-232C standard specifies certain characteristics of the signal that is sent through a modem. The first standard is *baud rate*. The term refers to the speed at which the modulated signal is sent through the communication lines.

The modems currently available for Commodore VIC 20 and Commodore 64 computers are 300-baud modems, which means that they transmit/receive data at about 30 characters per second. Fortunately, most bulletin boards, and the majority of computers with communications can converse in 300 baud.

The major disadvantage of the 300-baud rate is that it is relatively slow. The latest generation of modems transmits at 1200 baud, accomplishing data transfer in about 1/4 the amount of time that it takes for 300-baud communications.

If you were to hook up to a communications service/data utility such as CompuServe or The Source, you would have the choice of baud rates. Communicating at 1200 is probably preferable because, even though it is four times faster, it isn't four times as expensive. On the other side of the coin, however, the 1200 baud rate increases the probability of communications error, and may be prone to errors due to noise on the telephone lines. But in most cases, error is not a major problem.

In addition to baud rate, there are a number of other characteristics used by computers to format transferred data. When data is transferred, it can be done in a continuous stream, without waiting to see if the message has been received at the other end. If both computers operate in this mode, it is called *half duplex*. In half duplex communications, your computer can't really tell if the receiving computer is acting correctly.

With *full duplex* communications, however, your computer waits after each character for the receiving computer to "echo" the transmitted character. If the echo agrees with the originally transmitted character, the next character is sent, and so on until the entire transmission is completed.

When sending and receiving computers are not in the same duplex mode, strange things happen to the display. When your computer is in half duplex, for example, and the receiving computer is in full duplex, eevveerryy lleettteerr iiss pprriinntteedd ttwwiiccee.. By determining which form of duplex the receiving party is in and changing your settings accordingly, you can develop a configuration making communications possible.

In order to help the computer know when you are sending, or when you are waiting for it to start returning a message, a rather simple, but *absolutely essential* method to tell the two computers apart is also included in all modems (or can be a feature of your communications software). This feature is the selectable "originate/answer" mode. In most cases, when you are calling an information service, bulletin board or, for that matter, any other computer, your modem's switch should be set to "originate," since you are placing the call. The "answer"ing computer will listen for your computer's "originate" tone (which sounds different from its "answer" tone) and a positive connection will be made. Without the two different tones, the computers wouldn't be able to tell each other apart.

Parity and stop bits are also

terms which are applied to communications. Their definitions are probably beyond the range of this article. Suffice it to say that, once you have initially logged on to another computer, you will probably learn its normal baud rate, parity, stop bits and duplex requirements (and it will learn the capabilities of your computer) and both computers will rapidly learn to converse.

Once all the communications standards have been selected, you're just about ready to start to communicate. However, knowledge of standard settings, plus having a modem still aren't enough. Since computers are dumb devices, you will need a program that will let your computer make proper use of your hardware and the modem—a package known as communications software or a terminal emulator program.

What your communications software does is relatively simple. It tells your computer how to convert the computer's code into a form that can be sent through the modem, and how to decode data coming into your computer. In most cases, it allows your computer to act like a remote computer attached to a large mainframe computer—terminal emulation.

The least sophisticated program does just that—converts your computer into a "dumb" terminal. By dumb, we don't mean stupid —we mean that you won't be able to run programs or do any of the things that make a computer so useful to own. You'll be locked into the main computer's menus and can do only what it allows you to do. If you wanted to save your transactions on disk or tape, you would probably not be able to do it. You can only type text or instructions back and forth.

More sophisticated programs allow you to do more—much more. *SuperTerm* from Midwest Micro is an example of a full-featured, full-powered communications package for the Commodore 64. When *SuperTerm* loads into your 64, the first thing it asks you to do is set the time. While you are using the package, you can keep a running record of time that the program is in the computer, and also current time of day.

But that's only the beginning. If you were a writer, for example, you can transmit your story to another computer, reading it off your disk drive and sending it to a receiving computer. In the case of my imaginary "decorating" article, I sent it as electronic mail addressed to the magazine's private address on The Source. The editor of the magazine called The Source and "picked up" his mail. Neither call, incidentally, was a toll call although the letter was transmitted clear across country. The costs involved in sending the letter (hook-up time to The Source) was certainly much less than Express Mail, and transfer took only a few minutes.

In theory, a magazine may *never* have to print any of its author's articles onto paper (at least until it is ready to publish). What happens is this—all authors will be sending their articles electronically to the magazine. The magazine's editors read and edit the articles on their computer screens. When the article is polished into a form that meets their requirements, appropriate typesetting codes (type style, size of type, width of article, headlines, etc.) will be added to the text, and all this information will be transmitted by modem to the magazine's computerized typesetting equipment, where, on the first run, publication-ready copy will be produced. The savings in time and money using this system can be enormous.

Back to *SuperTerm*, however. With *SuperTerm* (and probably many other good communications programs) you can transfer data in both directions, storing your transactions onto a large "buffer" (memory inside the computer) and then onto disk. Entire programs, sales conferences, information or whatever else you can get from the other computer, can be saved onto your disk.

In combination with Midwest Micro's SmartASCII printer interface, you can "stream" your ongoing transactions onto a Centronics printer. "Stream" printing is printing while your communications are happening. This is a nice feature to have, since it's often difficult, if only inconvenient, to try to see what went on a few minutes earlier. Having hard copy to be able to scan through is a much appreciated feature.

SuperTerm also has a limited word processor, which allows you to enter and edit text. This processor is a primary means of transferring data or text, and is also the receive buffer. Therefore, if you want to send *and* receive, you will have to empty *your* text buffer before trying to record a message from the other end.

You can also load your password into memory, and simplify sign-on procedures for information services or bulletin boards. And, as with most good communications programs, you can easily alter the screen's graphics to provide you with a more readable display. (If you like red on green, you can have it). You can designate *your* messages in one color, and the *other* computer's messages in another color, so that, on the screen at least, you'll know exactly *who* said *what*.

We've only scratched the surface on telecommunications here. However, with a Commodore computer, a modem (one that automatically dials and answers the phone for you is now available), and appropriate terminal software, a whole new world is available to you. **C**

Mark Brownstein is a freelance writer and editor who works in Los Angeles and New York.

"Now Your Commodore 64™ Can Print Like a Pro!"

Grappler™ CD
Printer Interface

The Revolutionary Printer Interface for the Commodore 64™

A New Era in Commodore Printing Power.

Grappler CD offers the first complete answer to your printer interfacing requirements, with many powerful capabilities unique in the Commodore marketplace. Complete signal translation allows many popular name brand printers to operate perfectly with the Commodore 64, or to imitate Commodore's own printer. Even Commodore's graphic character set can be reproduced on Epson, Okidata, Star, ProWriter and other popular printers.

Exclusive Grappler CD features provide a variety of graphic screen dumps, text screen dumps and formatting. No other Commodore interface can offer this.

If you own a Commodore 64...

If you're serious about quality, trouble free printing... You need the Grappler CD.

Contact your nearest Commodore dealer or call Orange Micro for a dealer near you.

Commodore 64 and Commodore 1525 are trademarks of Commodore Electronics Limited.
Epson is a registered trademark of Epson America, Inc.

A Uniquely Intelligent Interface:

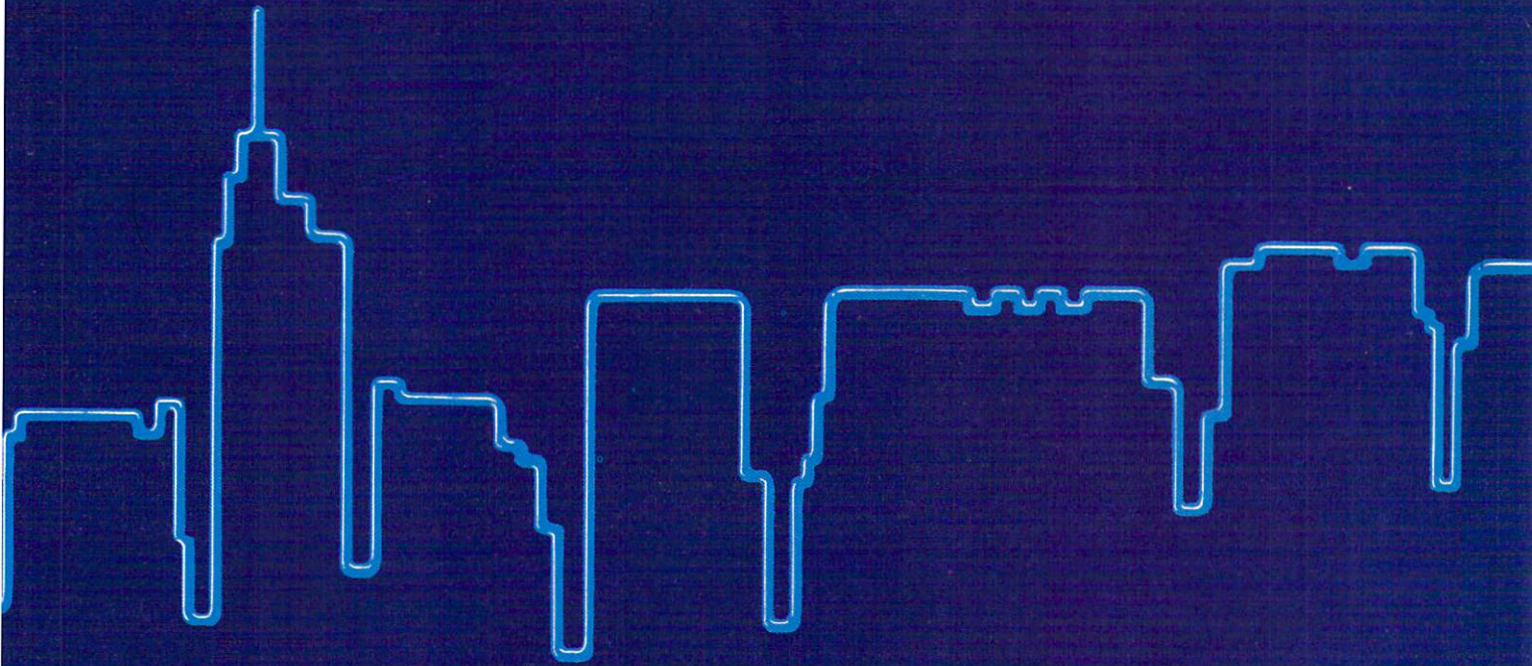
- Prints Screen Graphics Without Software
- Graphics Screen Dump Routines Include Rotated, Inversed, Enhanced and Double Sized Graphics.
- Full Code Translation From Commodore's PET ASCII to Standard ASCII, the Language of Most Printers.
- Complete Emulation of the Commodore 1525 Printer for printing of Commodore's Special Characters.
- Dip Switch Printer Selection for Epson, Star, Okidata, ProWriter and other popular printers.
- Conversion Mode for Easy Reading of Special Commodore Codes.
- Text Screen Dump and Formatting Commands
- 22 Unique Text and Graphics Commands

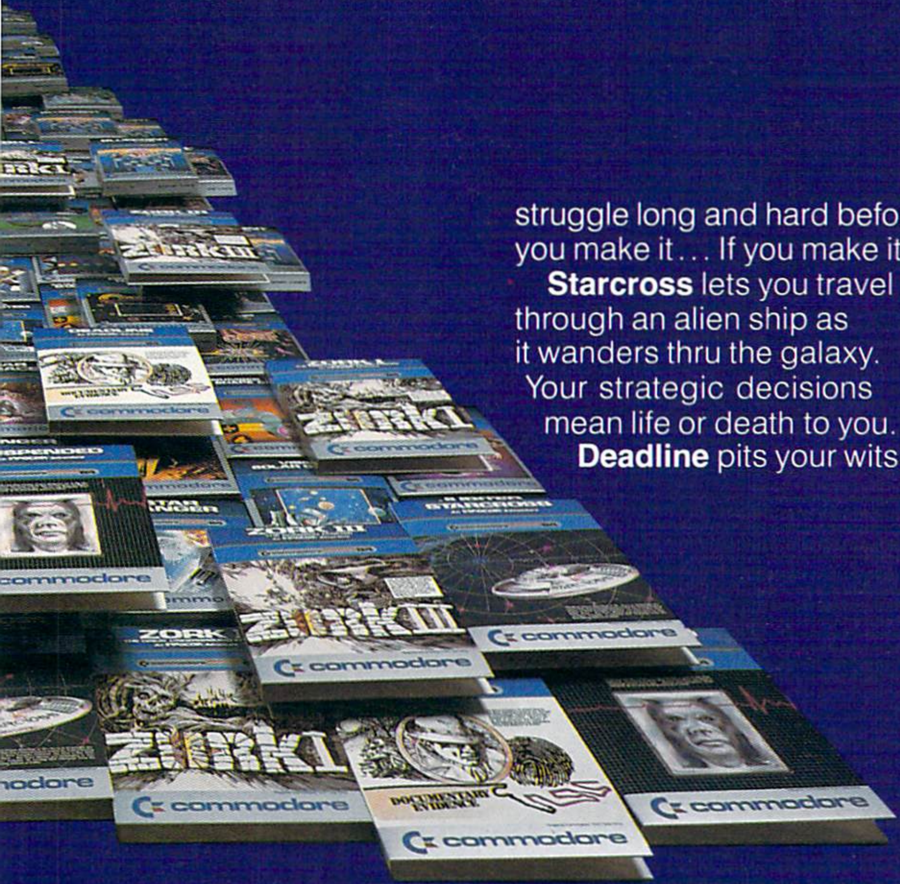
Orange Micro inc.

1400 N. LAKEVIEW AVE., ANAHEIM, CA 92807 U.S.A.
(714) 779-2772 TELEX: 183511CSMA

Circle Reader Service No. 12

The Best Adventure In Town





struggle long and hard before you make it... If you make it!

Starcross lets you travel through an alien ship as it wanders thru the galaxy. Your strategic decisions mean life or death to you.

Deadline pits your wits against a deadly murderer with time running out. Stretch your reasoning to the limit.

Suspended brings you to life 500 years into the future as the possible savior of your planet. Solve the riddles and you'll be a winner and a hero!

commodore
COMPUTERS

First In Quality Software

See your local dealer now... He's got the best game in town... just for you.

Pretty Printer

For the VIC 20 or Commodore 64 and 1525 Printer

by Robert Felice

Optimizing your BASIC programs so they run fast can also make the listings hard to read—unless you use this "pretty printer" to change your fast program back into a readable program listing.

BASIC is an interpreted computer language. Interpreted means that each program statement must be converted from high level BASIC into machine language each time it is executed by the computer. Therefore, it should not surprise us to find that the statement:

```
10FORI=1TO100:NEXT
```

will execute more quickly than the statement:

```
10 FOR I = 1 TO 100: NEXT
```

even though both statements instruct the computer to perform the same function. Why should this be? It is because the BASIC interpreter takes time to "step over" the blank characters.

Likewise, the statement:

```
10A=1:B=2:C=3
```

will execute faster than:

```
10A=1

20B=2

30C=3
```

Why? Because the BASIC interpreter takes time to "step over" the line numbers. Writing a program so that it runs quickly is called optimization. But whenever you optimize a BASIC program, it seems that you also destroy some of its readability.

There is a solution to this dilemma: a "pretty printer" program that can convert a highly optimized BASIC program into a readable program listing. Here is a functional summary of such a program:

1. All statement numbers are printed as right-justified five-digit numbers.
2. A space is inserted after all BASIC keywords.
3. REMarks are printed at the beginning of each statement. In addition, a REMark that begins with an equal sign is printed in double-width characters, and is preceded by a blank line. If the REMark begins with a non-blank character, it is also preceded by a blank line, but it is printed in normal-width characters.
4. Each line contains only one BASIC statement. The program fragment:

   ```
   10A=1:B=2:C=3
   ```

   is "pretty printed" as:

   ```
   10 A = 1:
      B = 2:
      C = 3
   ```

5. IF and FOR indent the succeeding lines of text, up to the end of the IF statement or up to the next NEXT. (Multiple nesting of IF's and FOR's is also supported).
6. Cursor control and color control characters are printed in brackets, e.g., "[clear]".

Implementing the Pretty Printer

In order to implement the pretty printer on your system, there are a few variables you might want to change:

1. PL, in line 50320, is the number of lines per page. It is currently set to 66, which is standard for 8.5 × 11 inch paper.
2. FL, in the same line, is the number of characters per line.
3. W$ and N$, in line 50360, are the printer commands for wide and normal printing, respectively. If you aren't using the VIC 1525 printer you may have to change these character strings.

Limitations

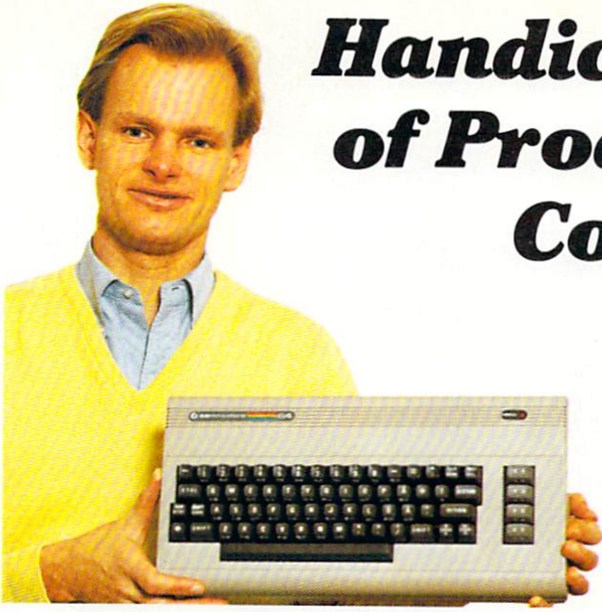
The program does have its limitations. First, it is slow. Each BASIC program statement typically takes between three and ten seconds to format. You can speed this up somewhat, by removing all the REMarks and renumbering the statements from one, but in so doing you will fall into the trap of optimizing at the expense of readability.

Secondly, a multiple NEXT such as:

```
10 FOR I = 1 TO 10
20 FOR J = 1 TO 10
30 NEXT J, I
```

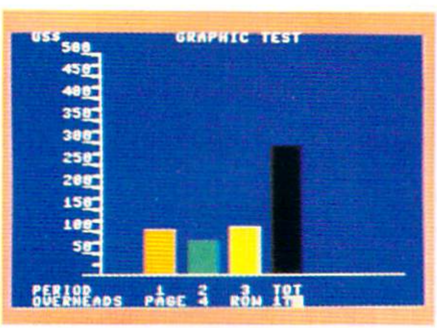
will not de-indent properly. Similarly, the construct:

Handic — A Complete Line of Products to Keep Your Commodore 64 Busy — Everyday!



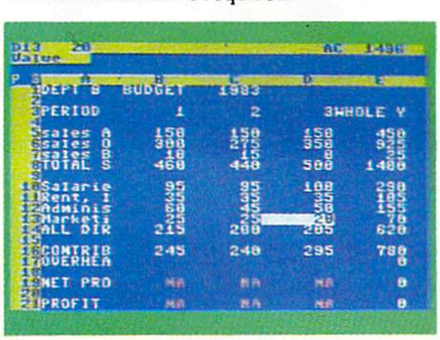
Handle your home budget, stock portfolio, loans and mortgages with Calc Result

Calc Result Easy is a simple-to-use spreadsheet program for the Commodore 64. It includes 254 lines × 64 columns, built-in graphics, and flexible printout formats. Plug-in cartridge... just plug it in and its ready. Perfect for cash flow analysis, personal net worth, IRA analysis, travel expenses, credit card expenditures, gas and electricity bills, etc.



Calc Result Easy $49.95

Calc Result Advanced gives you 32 pages of interrelated information. The three-dimensional feature allows you to consolidate calculations in summary format. Calc Result Advanced comes on plug-in cartridge and disk. Disk drive required.



Calc Result Advanced $99.95

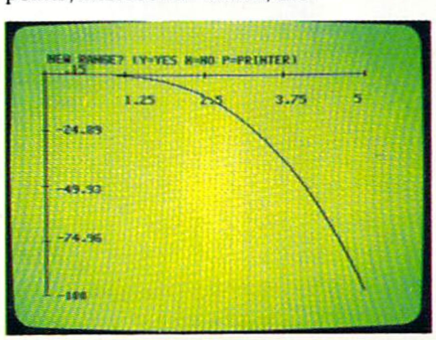
A complete database for the home

Addresses, telephone numbers, appointments, birthdays, or records–whatever you want to remember–put it on DIARY, an electronic notebook for home use. DIARY comes on a plug-in cartridge. It's easy to use and easy to learn, giving you the flexibility to design a personal calendar or address book.

Diary $29.95

Turn statistical information into graphic format

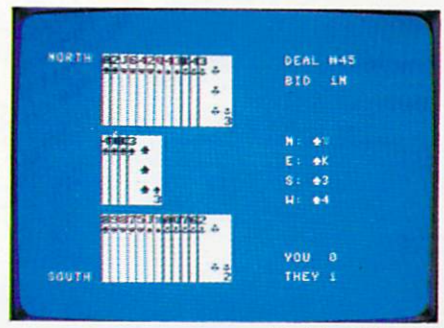
GRAF 64 converts mathematical functions into graphical analysis on the Commodore 64. An ideal program for studying math. Define a function, set the limits of an axis, plot a graph and display the extreme points, intersection values, etc.



Graf 64 $29.95

Develop your bridge skills

Whether you're an experienced bridge player or a beginner, polish your skills or learn the game with BRIDGE 64. Play North-South, then switch to East-West in the same deal, the return to that deal again and test your skill with a different strategy.



Bridge $39.95

Handic–for the broadest range of Commodore products

As the largest independent developer of Commodore software and accessories, Handic's broad range of business, education and recreation products are designed exclusively for the Commodore user who demands quality and reliability.

For more information and a catalogue of our products, see your nearest Commodore dealer, or call us direct.

handic software

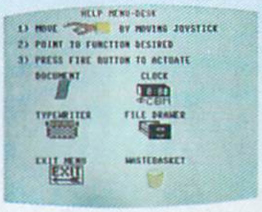
Handic Software, Inc.
Fellowship Business Center
520 Fellowship Road, B 206
Mount Laurel, NJ 08054
Phone (609) 663-0660

SOFTWARE SELECTION GUIDE

APPLICATION	SOFTWARE
Budget/Calculation	EASYCALC 64
Business Accounting	ACCOUNTS PAYABLE/CHECKWRITING, ACCOUNTS RECEIVABLE/BILLING, GENERAL LEDGER, INVENTORY MANAGEMENT, PAYROLL
Business Management	EASYFINANCE IV—BUSINESS MANAGEMENT
Children's Programming	ZORTEK & THE MICROCHIPS
Cooking/Recipes	MICRO COOKBOOK
Data Base Management	THE MANAGER
Electronic Spreadsheet	EASYCALC 64
Filing/Recordkeeping	MAGIC DESK, THE MANAGER, INVENTORY MANAGEMENT
Financial Investments	EASYFINANCE II—BASIC INVESTMENT ANALYSIS, EASYFINANCE III—ADVANCED INVESTMENT ANALYSIS, FINANCIAL ADVISOR
Graphics/Sound	SUPEREXPANDER 64
Learn Programming	INTRODUCTION TO BASIC—PART 1
Loans/Mortgages	EASYFINANCE I—LOAN ANALYSIS, FINANCIAL ADVISOR
Mailing List	EASYMAIL 64
Music	MUSIC COMPOSER, MUSIC MACHINE
Programming Aids	SUPEREXPANDER 64, SCREEN EDITOR, ASSEMBLER 64
Reference Books	PROGRAMMERS REFERENCE GUIDE, SOFTWARE ENCYCLOPEDIA
Spelling Dictionary	EASYSPELL 64 (for use with EASYSCRIPT 64)
Statistics/Forecasting	EASYFINANCE V—STATISTICS & FORECASTING, EASYFINANCE IV—BUSINESS MANAGEMENT
Teacher's Aids	EASYLESSON/EASYQUIZ, LOGO, PILOT
Telecommunications	VICMODEM, AUTOMODEM, TERM 20/64, RS232 INTERFACE
Wordprocessing	EASYSCRIPT 64, MAGIC DESK, WORD MACHINE/NAME MACHINE

MAGIC DESK I-TYPE & FILE

Only Commodore brings you the magic of MAGIC DESK... the next generation of "user-friendly" software! Imagine using your computer to type, file and edit personal letters and papers—without learning any special commands! All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go. MAGIC DESK is the "ultimate" in friendly software!



Special "Help" Menus

Not only is MAGIC DESK easy to use... it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next.

commodore
COMPUTERS

First In Quality Software

100 IF xxx THEN NEXT

will also not de-indent properly.

Finally, all excess spaces (including spaces within DATA statements) are stripped out by the pretty printer. Sometimes this is not a desirable effect.

Acknowledgments

The genesis of the pretty printer program was an article and listing that originally appeared in *Commodore Computing* magazine, published by Nick Hampshire in England. The author was listed only as CUP. I have transcribed the program to work with the VIC 20 and Commodore 64 operating systems, and the VIC 1525 printer. I have also fixed a few minor bugs that were in the original program and added the VIC 20 and Commodore 64 color control characters.

To give some idea of how useful "Pretty Printer" can be, I have run the program on itself! Even though the program is non-trivial, it is easy to read, especially the cursor control characters.

Should you wish a copy of the program without the hassle of having to type it in, I will make a copy of it on your cassette tape for a $3.00 fee. Send a blank cassette tape and a stamped, self-addressed return mailer to me at:

5902 Galaxie Road,
Garland, Texas 75042.

Pretty Printer

```
10000 REM=PRETTY PRINTER
10010 REM WRITTEN 11/03/80,CUP
10020 REM UPDATED 13/01/81,CUP
10030 REM UPDATED 13/7/83,ROBERT FELICE

10100 REM=DRIVER
10120 GOSUB 50000
10140 IF S<>0 THEN
          GOSUB 58000:
          END
10200 GOSUB 20000:
      IF S=0 THEN
          GOSUB 30000:
          GOSUB 33000:
          GOTO 10200
10300 GOSUB 58000:
      END

20000 REM=OBTAIN THE NEXT LINE

20100 REM--LINK AND LINE NR
      FOR EL=1 TO 4:
          GOSUB 21000:
          IF S=0 THEN
          NEXT EL:
20110 PRINT CR$"line"CL(4)*256+CL(3)

20200 REM--REST OF LINE
20220 IF S=0 THEN
          FOR EL=5 TO 255:
              GOSUB 21000:
              IF CL(EL) THEN
              NEXT EL
20999 RETURN

21000 REM=GET THE NEXT CHARACTER
```

```
21020 GET #1,C$:
      CL(EL)=ASC(C$+Z$):
      S=ST:
      RETURN
21999 RETURN

30000 REM=PRINT COMMENTS
30020 Q=0:
      SM=LM:
      EN=0

30030 REM--HUNT FOR COMMENT
30040 FOR CO=5 TO EL
30050     Q=(CL(CO)=QU)=(NOT Q):
          IF NOT Q THEN
              IF CL(CO)=143 GOTO 30100
30060 NEXT CO:
      GOSUB 32000:
      RETURN

30100 REM--BLANK LINE REQUIRED?
30120 B=CO:
      T=CL(B+1):
      IF T<>0 AND T<>32 THEN
          P$="":
          GOSUB 40000
30140 GOSUB 32000:
      C=CL(B):
      TK=-1:
      C7=15:
      GOSUB 34000

30200 REM--ENHANCED MODE REQUIRED?
30220 EN=(T=61):
      IF EN THEN
          P$=P$+W$
30240 BE=CO+1-EN:
      IF BE>=EL GOTO 30400

30300 REM=PRINT COMMENTS
30320 FOR CO=BE TO EL-1:
          GOSUB 35000:
          GOSUB 34000:
      NEXT CO

30400 REM--CANCEL ENHNCED MODE
      IF EN THEN
          P$=P$+N$:

30500 REM--MARK NEW EOL
      CL(B)=0:
      EL=B:
      LL=FL+EN*HL:
```

```
       GOSUB 40000:
       LL=FL:

31000 REM=SET UP FOR NEXT LINE
31020 P$=LEFT$(B$,SM):
       RETURN

32000 REM=LINE NR SETUP
32040 P$=LEFT$(RIGHT$(B$+STR$(CL(3)+CL(4)*256),5)+B$,SM):
       RETURN

33000 REM=REST OF LINE
33020 CO=5:
       Q=0

33100 REM--NEXT STATEMENT
       BE=CO:
       IF CO=EL THEN
          RETURN:
33200 REM CONTENTS OF TOKEN
       GOSUB 35000:

33240 REM--LEADING SPACE REQUIRED?
33260 IF CO<>BE AND NOT Q THEN
          IF TK THEN
             IF C$="p" OR C$="b" THEN
                P$=P$+" "

33300 REM--DE-INDENTATICINS REQUIRED?
33320 REM NEXT?
       IF C=130 AND NOT Q THEN
          SM=SM-3:
          LM=LM-3:
          P$=LEFT$(P$,SM):

33400 REM--DECODE IF NOT SPACE
       IF Q OR (C<>32 AND NOT Q) THEN
          GOSUB 34000:

33500 REM--TRAILING SPACE REQUIRED?
33520 IF TK AND NOT Q THEN
          IF C$="s" OR C$="b" THEN
             P$=P$+" "

33600 REM--INDENTATION REQUIRED
33620 REM IF?
       IF C=139 THEN
          SM=SM+3:
33640 REM FOR?
       IF C=129 THEN
          SM=SM+3:
          LM=LM+3:

33700 REM--NEXT TOKEN
33720 CO=CO+1:
```

```
      IF CO<EL THEN
         IF (C<>167 AND C<>58) OR Q GOTO 33200
33740 GOSUB 40000:
      GOSUB 31000:
      IF CO<EL GOTO 33100

34000 REM==DECODER
34020 IF Q GOTO 34100
34040 REM TOKEN
      IF TK THEN
         P$=P$+MID$(T$(C7),2):
         RETURN:
34060 REM FORCE UPPER CASE
      IF C>64 AND C<91 THEN
         P$=P$+CHR$(C OR 32):
         RETURN:

34100 REM--STRINGS
34120 REM CONTROL CHAR?
34140 FOR J=1 TO NC:
          IF C=CC(J) THEN
             P$=P$+"["+CC$(J)+"]":
             RETURN

34160     IF C>CC(J) THEN
         NEXT J
34180 P$=P$+CHR$(C):
      RETURN

35000 REM==NEXT  TOKEN
35020 C=CL(CO):
      Q=(C=QU)=(NOT Q):
      TK=C>=128 AND NT>=C AND NOT Q
35040 IF TK THEN
         C7=C AND 127:
         C$=LEFT$(T$(C7),1)
35999 RETURN

40000 REM==OUTPUT  GENERATOR

40100 REM--IS LINE SHORT ENOUGH?
40120 IF LEN(P$)<=LL THEN
         TM$=P$:
         GOSUB 41000:
         RETURN
40140 T$=LEFT$(B$,SM):
      IF EN THEN
         T$=T$+W$

40200 REM--SEARCH FOR SPACE
40220 FOR I=LL+1 TO HL STEP -1:
         IF MID$(P$,I,1)<>" " THEN
         NEXT I:
         I=LL+1
```

```
40240 TM$=LEFT$(P$,I-1):
      IF EN THEN
         TM$=TM$+N$
40260 IF MID$(P$,I,1)=" " THEN
         I=I+1
40280 P$=T$+MID$(P$,I):
      GOSUB 41000:
      GOTO 40100

41000 REM=PRINTER CONTROL
41020 IF CL<>PL GOTO 41200

41040 REM--NEW PAGE
      CL=0:
      PG=PG+1:
41060 X$="":
      FOR J=1 TO 3:
         GOSUB 43000:
      NEXT J:
      X$=TL$+STR$(PG):
      GOSUB 43000
41080 X$="":
      FOR J=1 TO 2:
         GOSUB 43000:
      NEXT J

41200 REM--TRANSMIT
      X$=TM$:
      GOSUB 43000:

41300 REM--BOTTOM OF FORM?
41320 IF CL<BF THEN
         RETURN

42000 REM=FORM FEED
42020 X$=""
42100 IF CL<>PL THEN
          GOSUB 43000:
         GOTO 42100
42999 RETURN

43000 REM=XMIT LINE TO PRINTER
43040 PRINT #P,X$:
      CL=CL+1:
      RETURN

50000 REM=INITIALIZATON
50020 CLR:
      PRINT CHR$(14)"[clear][down][down]basic Pretty Printer"
50200 X$=""

50300 REM--CONSTANTS
50320 D=4:
      FL=80:
```

```
      NC=40:
      NT=218:
      P=4:
      PL=66:
      QU=34
50340 BF=PL-4:
      HL=INT(FL/2)
50360 CR$=CHR$(13):
      N$=CHR$(15):
      W$=CHR$(14):
      Z$=CHR$(0)
50380 REM MAX NESTING IS 20
      B$=" ":
      FOR I=1 TO 6:
          B$=B$+B$:
      NEXT I:
50400 REM INITIALIZED VARIABLES
50410 CL=PL:
      LL=FL:
      LM=6:
      PG=0:
      RT=0

50420 REM--VARIABLES
50430 BE=C<CO<C7<D<EL<EM<EN<ET<ES<I<J<Q<S<SM<T<TK
50440 C$=X$:
      D$=X$:
      F$=X$:
      P$=X$:
      T$=X$:
      TL$=X$:
      TM$=X$

50500 REM--ARRAYS
50520 DIM T$(127 AND NT):
      DIM CC(NC):
      DIM CC$(NC):
      DIM CL(255):
      DIM TR(7)

50600 REM--FILES
50620 PRINT "[down]Filename:"
50630 PRINT "  "CHR$(QU)"                    "CHR$(QU)CR$"[up]";:
      INPUT F$
50632 F$=F$+" "
50634 FOR I=16 TO 1 STEP -1:
          IF MID$(F$,I,1)=" " THEN
          NEXT I
50636 F$=LEFT$(F$,I)
50640 PRINT "[down]Date:"CR$"  "CHR$(QU)"  /  /  "CHR$(QU)CR$"
      [up]";:INPUT D$
50660 PRINT "Time:"CR$"  "CHR$(QU)"  :  "CHR$(QU)CR$"[up]";:
      INPUT T$
```

```
50680 TL$=LEFT$("Listing of "+F$+" on "+D$+" at "+T$+B$,70)+"Page"
50700 OPEN P,D,7

50800 REM--TAGS ETC
      PRINT "[down][rvs]Initializing[off]":
50820 FOR I=0 TO 127 AND NT:
         READ T$(I):
      NEXT I
50840 FOR I=1 TO NC:
         READ CC(I):
         READ CC$(I):
      NEXT I
50880 OPEN 5,8,15:
      OPEN 1,8,2,"0:"+F$+",p,r":
      GOSUB 50990
50900 REM SKIP OVER LOAD ADDR
      GET #1,C$:
      GET #1,C$:
50980 REM RETURN ADDR CLEARED BY CLR
      GOTO 10140:
50990 S=ST:
      INPUT #5,EM,ER$,ET,ES:
      IF EM>0 THEN
         PRINT "disc error nr:"EM;CR$;ER$;XR$;ET;ES
50995 RETURN

51000 REM=TOKENS
51125 DATA "nEND","sFOR"
51130 DATA "sNEXT","sDATA","nINPUT #","sINPUT","sDIM"
51135 DATA "sREAD","sLET","bGOTO","sRUN","sIF"
51140 DATA "nRESTORE","bGOSUB","nRETURN","nREM","sSTOP"
51145 DATA "sON","sWAIT","sLOAD","sSAVE","sVERIFY"
51150 DATA "sDEF","sPOKE","nPRINT #","sPRINT","nCONT"
51155 DATA "sLIST","nCLR","sCMD","sSYS","sOPEN"
51160 DATA "sCLOSE","sGET","nNEW","nTAB(","bTO"
51165 DATA "nFN","nSPC(","pTHEN","sNOT","bSTEP"
51170 DATA "n+","n-","n*","n/","n↑"
51175 DATA "bAND","bOR","n>","n=","n<"
51180 DATA "nSGN","nINT","nABS","nUSR","sFRE"
51185 DATA "nPOS","nSQR","nRND","nLOG","nEXP"
51190 DATA "nCOS","nSIN","nTAN","nATN","nPEEK"
51195 DATA "nLEN","nSTR$","nVAL","nASC","nCHR$"
51200 DATA "nLEFT$","nRIGHT$","nMID$","bGO","sCONCAT"
51205 DATA "sDOPEN","sDCLOSE","sRECORD","sHEADER","sCOLLECT"
51210 DATA "sBACKUP","sCOPY","sAPPEND","sDSAVE","sDLOAD"
51215 DATA "sCATALOG","sRENAME","sSCRATCH","sDIRECTORY"

52000 REM=CURSOR  CONTROL
52010 DATA 005,"wht"
52020 DATA 013,"ret"
52030 DATA 014,"lc"
52040 DATA 017,"down"
52050 DATA 018,"rvs"
52060 DATA 019,"home"
```

```
52070 DATA 020,"del"
52080 DATA 028,"red"
52090 DATA 029,"right"
52100 DATA 030,"grn"
52110 DATA 031,"blu"
52120 DATA 129,"orng"
52130 DATA 131,"run"
52140 DATA 133,"f1"
52150 DATA 134,"f3"
52160 DATA 135,"f5"
52170 DATA 136,"f7"
52180 DATA 137,"f2"
52190 DATA 138,"f4"
52200 DATA 139,"f6"
52210 DATA 140,"f8"
52220 DATA 141,"shf ret"
52230 DATA 142,"uc"
52240 DATA 144,"blk"
52250 DATA 145,"up"
52260 DATA 146,"off"
52270 DATA 147,"clear"
52280 DATA 148,"inst"
52290 DATA 149,"brn"
52300 DATA 150,"lt red"
52310 DATA 151,"grey1"
52320 DATA 152,"grey2"
52330 DATA 153,"lt grn"
52340 DATA 154,"lt blu"
52350 DATA 155,"grey3"
52360 DATA 156,"pur"
52370 DATA 157,"left"
52380 DATA 158,"yel"
52390 DATA 159,"cyn"
52400 DATA 225,"pi"

58000 REM═TERMINATE
58010 GOSUB 42000:
      CLOSE 1:
      CLOSE 15:
      CLOSE P:
      END
```

C

WE LOVE COMMODORE
and
We Love Our Customers

That's why we only sell and support Commodore 64 and Vic 20 computers!! We have • the best prices • over 1000 programs • 500 accessories • absolutely the best service • one day express mail delivery • immediate replacement warranty • 15 day free trial • programming knowledge • technical knowledge • we are the only one in the U.S.A. with complete support for Commodore 64 and Vic 20 computers!!

PROTECTO ENTERPRIZES Box 550, Barrington, IL 60010

SEND IN THIS COUPON TODAY FOR A FREE 64-PAGE "EXCLUSIVE COMMODORE-64 AND VIC-20 CATALOG" — PLUS OUR SPECIAL

"BUY MORE—SAVE MORE COUPON"

(Save up to $500 on software and accessories)

Name ______________________________

Address ______________________________

City __________ State __________ Zip Code __________

No One! But No One! Can Compare

TO

PROTECTO ENTERPRIZES

TO ORDER WRITE OR CALL: PROTECTO ENTERPRIZES, BOX 550, BARRINGTON, IL 60010

Call 312/ 382-5244

8 to 5 Weekdays 9-12 Saturdays

(See Next 4 Pages)

COMMODORE 64

(more power than Apple II at half the price)

$99.50*

- 170K DISK DRIVE $159.00*
- TRACTION FRICTION PRINTER $79.00*

COMPUTER AND SOFTWARE SALE

WE HAVE THE BEST SERVICE

WE HAVE THE LOWEST PRICES

VIC-20

(a real computer at the price of a toy)

$79.50

- COM-64 POWER FOR VIC-20 $79.00
- NEW VOICE SYNTHESIZER $59.00 (Com-64 or VIC-20)

* COMMODORE 64 COMPUTER $99.50

You pay only $199.50 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your computer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net computer cost is $99.50!!

*170K DISK DRIVE $159.00

You pay only $259.00 when you order the 170K Disk Drive! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your disk drive that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net disk drive cost is $159.00.

* TRACTION FRICTION PRINTER $79.00

You pay only $179.00 when you order the Comstar T/F deluxe line printer that prints 8 1/2 x 11 full size, single sheet, roll or fan fold paper, labels etc. 40, 66, 80, 132 columns. Impact dot matrix, bi-directional, 80 CPS. LESS the value of the SPECIAL SOFTWARE COUPON we pack with your printer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied your net printer cost is only $79.00.

4 COLOR PRINTER/PLOTTER $99.00

Lowest cost, 4 color, 80 column, letter quality PRINTER/PLOTTER for Com-64 or VIC-20 computers!! List programs. High resolution graphics for charts and geometric figures. INCLUDES INTERFACE AND SPECIAL SOFTWARE SAVINGS COUPON!!

80 COLUMN BOARD $99.00

Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! PLUS—you can get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET. List $59.00 **SALE $24.95** if purchased with 80 COLUMN BOARD!! (Tape or Disk)

80 COLUMNS IN COLOR EXECUTIVE WORD PROCESSOR $69.00

This EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE for PROFESSIONAL Word-processing application! DISPLAYS 40 OR 80 COLUMNS IN COLOR or Black and White! Simple to operate, powerful text editing with a 250 WORD DICTIONARY, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! Includes a powerful mail merge. 20,000 WORD DICTIONARY - List $24.95 **SALE $19.95.** EXECUTIVE DATA BASE - List $69.00 **SALE $49.00.** (Disk

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE COUPON with every COMMODORE 64 COMPUTER-DISK DRIVE-PRINTER-MONITOR we sell! This coupon allows you to SAVE OVER $100 OFF SALE PRICES! Up to $500 savings are possible!!

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
Executive Word Processor	$99.00	$69.00	$59.00
Executive Data Base	$69.00	$59.00	$39.00
20,000 Word Dictionary	$24.95	$19.95	$14.95
Electronic Spreadsheet	$59.95	$49.00	$39.00
Accounting Pack	$49.00	$39.00	$29.00
Total 5.2 Word Processor			
Tape	$69.00	$49.00	$34.00
Disk	$79.95	$59.00	$39.00
Total Text 2.6 Word Processor			
Tape	$44.95	$34.95	$22.00
Disk	$49.00	$39.00	$27.00
Total Label 2.6			
Tape	$24.95	$18.00	$12.00
Disk	$29.95	$23.00	$15.00
Programmers Helper (Disk)	$59.00	$39.95	$29.95
80 Column Screen (Disk)	$59.95	$39.95	$29.95
Crush-Crumble-Chomp (Tape/Disk)	$29.95	$24.95	$19.95
Pitstop (Cartridge)	$39.95	$29.95	$24.95
Typing Teacher (Tape/Disk)	$29.95	$24.95	$15.00
Sprite Designer (Disk)	$16.95	$14.95	$10.00
Fireball Joy Stick	$24.95	$15.95	$10.00
Light Pen	$39.95	$16.95	$14.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60

(See 100 coupon items in our catalog!)

Write or call for Sample SPECIAL SOFTWARE COUPON!

EXECUTIVE QUALITY PROFESSIONAL BUSINESS SOFTWARE

The Cadillac of business programs for Commodore 64 Computers

Item	List	*SALE	Coupon
Inventory Management	$99.00	$59.00	$49.00
Accounts Receivable	$99.00	$59.00	$49.00
Accounts Payable	$99.00	$59.00	$49.00
Payroll	$99.00	$59.00	$49.00
General Ledger	$99.00	$59.00	$49.00

VIC-20 COMPUTER $79.50

This 25K VIC-20 computer includes a full size 66 key typewriter keyboard color and graphics keys, upper/lower case, full screen editor, 16K level II microsoft basic, sound and music, real time floating point decimal, self teaching book, connects to any T.V. or monitor!

COM-64 POWER FOR VIC-20 $79.00

Just plug in our 32K RAM MEMORY EXPANDER and you get as much usable programming power as the Commodore-64 computer!! Master control switches on cover, Gold Edge connectors, five year warranty (FREE $29.95; CARTRIDGE GAME)

NEW VOICE SYNTHESIZER $59.00

For Com-64 or VIC-20 computers. Just plug it in and you can program words and sentences, adjust volume and pitch, make talking adventure games, sound action games and customized talkies!! FOR ONLY $19.95 you can add TEXT TO SPEECH, just type a word and hear your computer talk—ADD SOUND TO "ZORK," SCOTT ADAMS AND AARDVARK ADVENTURE GAMES!! (Disk or tape).

16K RAM CARTRIDGE $49.00

Increases VIC-20 programming power 4 times. Expands total memory to 41K (41,000 bytes). Memory block switches are an outside cover! CARDCO Includes FREE $29.95 game!!

8K RAM CARTRIDGE $34.95

Increases VIC-20 programming power 2 1/2 times. Expands total memory to 33K (33,000 bytes). Includes FREE $16.95 game.

3 SLOT SWITCHABLE EXPANDER $24.95

Allows you to add 3 cartridges at one time—switch select to turn slots on or off—PLUS reset button. A must for your VIC-20 computer!!

60K MEMORY EXPANDER $49.00

Sixslot Board — Switch selectable — Reset button — Ribbon cable — CARDCO. A must to get the most out of your VIC-20 Computer!

9" GREEN SCREEN MONITOR $69.00

Excellent quality SANYO, easy to read, 80 columns x 24 lines, Green Phosphorous screen with anti-glare, metal cabinet! Saves your T.V. PLUS $9.95 for connecting cable. Com-64 or VIC-20.

12" GREEN OR AMBER MONITOR $99.00

Your choice of green or amber screen monitor, top quality, SANYO, 80 columns x 24 lines, easy to ready, anti-glare, faster scanning! A must for word processing PLUS $9.95 for connecting cable. Com-64 or VIC-20.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD — C.O.D.

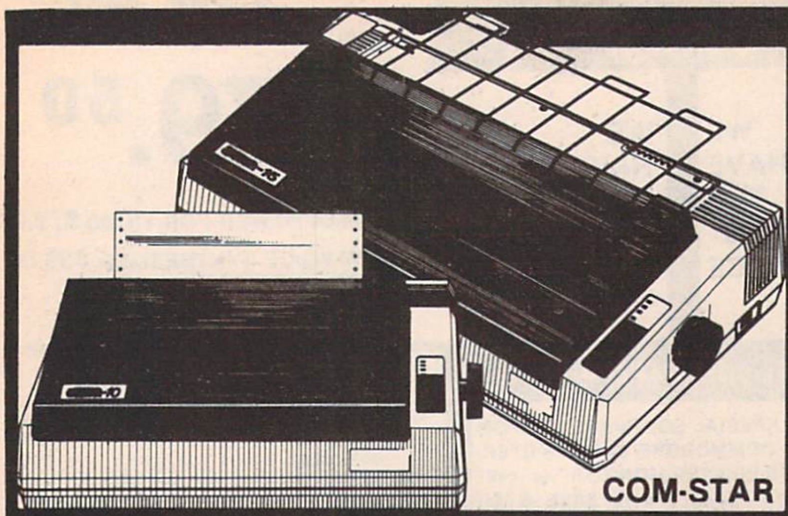
PROTECTO ENTERPRIZES

(WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Circle Reader Service No. 15

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor Friction Printer

only $ 179**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second • 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing • Print Labels, Letters, Graphs and Tables • List Your Programs
- Print Out Data from Modem Services • "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$179.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$399.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$529.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64—$69.00
For Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Circle Reader Service No. 15

Random Thoughts

Part 6: A Walk on the Wild Side

by Mark Zimmermann

In months past we've looked at how to make and use a variety of random number distributions on your computer. This time, we're going to investigate a random process called the "random walk". It shows up just about everywhere, from the drifting of perfume through the air to a gambler's misfortune at the roulette table. Some of the tools we encountered in the past will prove useful again as we analyze the random walk in detail.

What is a Random Walk?

Some years ago, when alcoholism wasn't viewed as a disease, drunks seemed more amusing than they do to us today. A drunk meandering down the street, staggering this way and that, performed what was called a "drunkard's walk". Each step was in a new direction, unrelated to the previous step.

Mathematicians saw the drunkard's walk not as a tragicomedy but as a fascinating process to be studied in the abstract. They called the process a random walk. Its key feature is the repetition, independently, of a large number of individual random steps. If you look at a random walk locally, in sharp enough detail to see the steps, it seems totally chaotic. But step back and look at it on a larger scale! Then a lot of the fluctuations cancel out, and you can see the random walk as a smooth, gradual drift which obeys some simple rules. But, although the rules are simple, they're not always obvious.

The Rules of the Walk

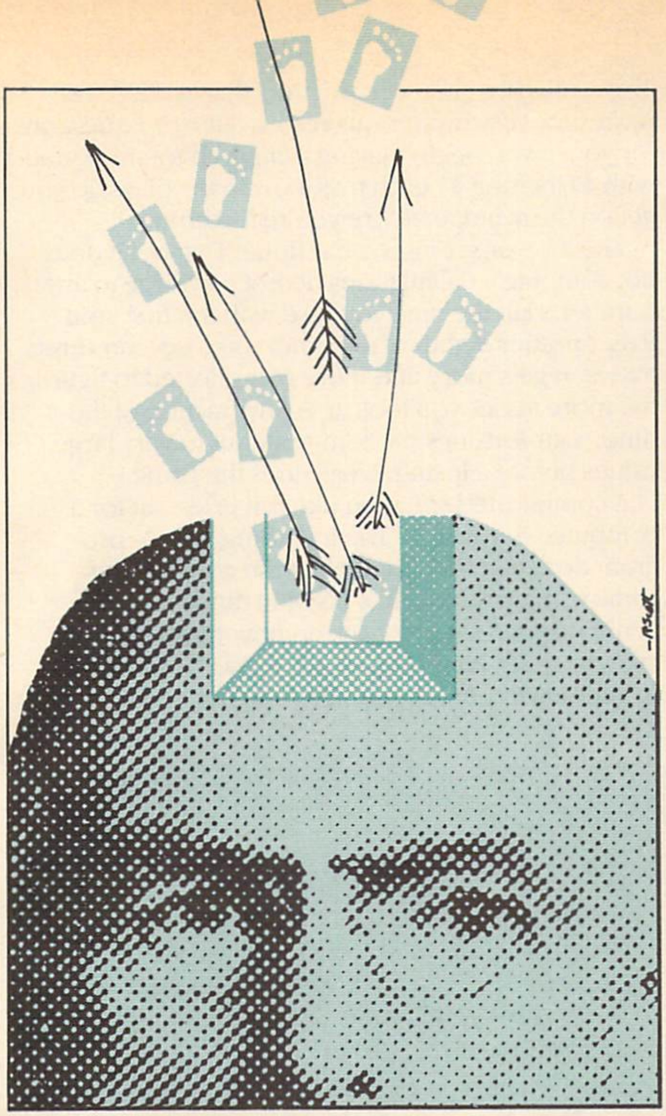
The first of the rules that govern random walks is an old friend of ours: the "square root of N" rule-of-thumb. After N steps, how far is the random walk away from where it started? Answer: on the average, SQR(N), the square root of N, times the length of a single step! This makes sense. Consider the motion in one dimension, say left vs. right. We can imagine tossing a coin at each step, and moving left if we get a head and right if we get a tail. After a large number N tosses of the coin, we will have gotten T tails and N minus T heads. So, we will have gone T steps right, N minus T steps left, and will have ended up T−(N−T) = 2*T−N steps to the right of where we started.

On the average, if we have a fair coin, we get tails half the time. That is, T=N/2. Our random walk, therefore, on the average ends up where we started. Isn't that a contradiction? No! All it says is that we are equally likely to be left or right of our starting point. If we count distances to the right as positive numbers, and distances to the left as negative, then of course the average is zero. That doesn't answer the question we asked. How far, on the average, are we away?

To get the average distance we have moved, we must figure out some method of averaging that avoids canceling out positive and negative numbers. A convenient choice is our friend from previous articles, the root-mean-square or RMS distance. Do 100 walks, each of length N steps. Take the final result of each walk as a number; it will lie between −N and +N. Square those numbers, add up the squares, divide by 100 to get the mean square and take the square root of that, the "root-mean-square".

Here's a simple BASIC program you can run to do just what we've described:

```
10 A=0
20 FOR I=1 TO 100: REM DO IT 100 TIMES
30 X=0: REM START OUT AT ORIGIN
   EVERY TIME
40 FOR J=1 TO 25: REM TAKE 25 STEPS
   (N=25)
50 R=RND(1): S=-1: IF R>.5 THEN S=1:
   REM S IS THE STEP: LEFT=-1, RIGHT=+1
60 X=X+S
70 NEXT J
80 PRINT X
90 A=A+X*X: REM ACCUMULATE X SQUARED
    IN A
100 NEXT I
110 PRINT "RMS DISTANCE = ";SQR(A/100)
    : REM DISPLAY RESULT
    120 END
```



If you try it, you'll find that the average fluctuates but tends to be pretty close to five, the square root of 25, our choice for N. You may want to change the choice of N; try 36 or 49, for example. If you increase the number of runs, you'll get more accurate answers. (Think about how accurate you expect averaging 100 runs to be; you can apply the square-root-of-N rule of thumb to that, if you look at it right!)

More Rules of the Random Walk

What if the random walk takes place in more than one dimension? For instance, suppose you're thinking about a bug randomly crawling inside a bathtub, and want to know how long it will take to fall down the drain. (That may be a tough question to answer, since the bug bounces off the walls and the walk isn't a simple random one.) You have a two-dimensional random walk then: the bug can go north-south, and it can go east-west. Or suppose you are studying a dust particle drifting about in the air; that's a three-dimensional walk.

The rule in more than one dimension is simple: break the problem up and look at each dimension separately. If the walk is truly random, so that the direction of motion at each step is completely independent of the previous step, and the motion in one dimension is independent of the motion in any other dimension, then you can analyze each dimension as a simple one-dimensional walk. At the end, you can put the dimensions back together to get the answer.

For example, suppose a husband and wife are playing roulette in a casino. They each start with $100, and bet $1 on every spin of the wheel. The man bets on red or black, while the woman bets on odd or even. Suppose also that the roulette wheel is fair and (highly unlikely) has no green 0 or 00 slots to give the house an edge.

Then what we have here is a perfect two-dimensional random walk in which the money held by each member of the couple wanders independently (until one or both go bankrupt or the casino throws them out). Each individual's fortune averages $100, the starting amount, and after N plays the RMS distance each person's money supply has drifted is just SQR(N). If you want to know the distribution of the couple's total fortune, just add the separate distributions.

A bug (or a drunkard) that randomly takes a step of length one in a random direction gives only a slightly more complicated situation. Look again at each dimension separately. The steps in the north-south dimension are random, independent, with average value zero but with a nonzero RMS step size. The same is true for the steps in the east-west dimensions. By symmetry, the RMS step size has to equal the same thing in both dimensions. The Pythagorean theorem says that the square of the total step length (which we said was one) must equal the sum of the squares of the steps in each dimension. So, we can deduce that the RMS step length in each dimension is just SQR(1/2) = 0.7071…

Given the RMS step size, we can apply our first rule (SQR(N)) to tell us that, on the average, after N steps we are SQR(N)*SQR(1/2) away from where we started in the north-south dimension, and the same in the east-west dimension. Again applying the Pythagorean theorem, we see that the net average distance the two-dimensional random walk has moved after N steps is just SQR((SQR(N)*SQR (1/2))↑2 + (SQR(N)*SQR(1/2))↑2) = SQR(N). Quite a simple result!

If you aren't sure that the above deductions are correct, try running a simulation on your computer to check. (I must confess that I had to check it myself.) This simple program generates a step of length one in a random direction, adds up 25 of them, and sees how far we are from the origin then. It averages 100 of those random walks and prints out the answer.

```
10 A=0: FOR I=1 TO 100
20 X=0: Y=0: PP=2*3.1415926
```

```
30 FOR J=1 TO 25: R=RND(1)*PP:
   X=X+SIN(R): Y=Y+COS(R): NEXT J
40 PRINT "WALK #";I;" ENDED AT X=";X;
   ", Y=";Y
50 DD=X*X+Y*Y: PRINT "DISTANCE FROM
   START=";SQR(DD)
60 A=A+DD:NEXT I
70 PRINT: PRINT "RMS DISTANCE=";SQR
   (A/100)
80 END
```

If you try it, you'll discover that the average distance is about five units, the square root of the 25 steps we took. (It may take a couple of minutes to run; I left out the remarks and squeezed the inner loop all onto line 30 to speed it up a bit.)

More General Walks

All the walks we've looked at so far have been "fair"—equally likely to go in any direction. Suppose that there's a bias? For instance, in a real roulette game the odds are 2/38ths against the player (about 5%). How does that change the random walk that a gambler's fortune takes?

The answer is simple: if each step has a bias, then the expected final outcome drifts with that bias every step. If the gambler has a chance 18/38 to win $1 and 20/38 to lose $1, then every bet has a net drift of −$0.0526.... After N steps, the expected value to end up at is N times the drift per step. There will still be fluctuations about that expected outcome, with size proportional to SQR(N). But instead of an RMS variation of SQR(N) that we got for the fair game, the spread will be a bit smaller. It's smaller because, when you work out the RMS step size relative to the drift, that RMS step is a bit less than one.

An extreme case may make that last point clearer. Suppose the game is *very* unfair, so that the player loses every time. Then the drift is −1, and after N steps the player has always lost N dollars. There is no RMS fluctuation about the average, because relative to the drift the steps don't go anywhere. Think about it. In a less extreme case, if the player loses 99% of the time, you can see that there still is very little RMS fluctuation about the average drift value. So although the spread in the expected outcome about the average still grows like a constant times SQR(N), that constant is very small. Do some numerical experiments and see how it develops, if you're curious.

A Final Mystery

I don't want to leave the impression that everything about the random walk has a simple answer. The walks are unbounded, and that sometimes makes for very mysterious results. Here's a question for you: if you begin playing a fair coin tossing game with $1, betting $1 each toss, how many plays do you get on the average before you're bankrupt?

The answer seems to be infinite! That is, "it does not compute". Calculations do not converge to any number. Half the time you lose with the first coin toss. Another eighth of the time, you lose after three tosses. It gets more and more complicated to figure, the more tosses you look at. A tiny fraction of the time, your fortunes random-walk out to very large values before a losing streak ends the game.

A complicated situation like this cries out for a computer simulation. The following simple program does that for you. It plays the coin tossing game using your BASIC's RND(1) random number generator and reports back on how many steps it takes to reach bankruptcy each game, in addition to keeping a running average.

```
10 A=0: N=1
20 X=1: T=0
30 T=T+1: R=RND(1): S=1: IF R>0.5
   THEN S=-1
40 X=X+S: IF X>0 GOTO 30
50 A=A+T: PRINT "BANKRUPT AFTER";T;"
   STEPS--AVERAGE=";A/N
60 N=N+1: GOTO 20
```

When you start the program, you'll probably get a number of fairly short runs, giving an average of only five or ten. Then a lucky run of a few hundred steps will show up and pull that average up. After a while longer, a still luckier run will come with a few thousand or more steps, to pull the average up still higher... and so forth. The longer you run the program, the higher the average will tend to drift. The only limitations on how high it will go are the precision of the RND(1) function and the time you have available to run the program.

There are other mysteries about the random walk. In one dimension, for instance, a simple walk will revisit its starting point an infinite number of times. In two dimensions, I am told (but have not proved) that a walk will come back arbitrarily close to where it began, if you wait long enough. That is, if you want to get within a distance .000001 of where you started, you can do it, but it may take trillions of steps. I don't know what happens in three or more dimensions. That's a subject for you (and your computer) to explore! **C**

Mark Zimmerman is a regular contributor to Commodore Microcomputers.

To Round or Not to Round

by Richard H. Goodyear, Ph.D.

Your Commodore 64 or VIC 20 may be math wizards— except when it comes to rounding certain numbers.

The process of rounding numbers is one that must be done during and/or at the end of many types of calculations. If not done at the proper steps in a series of calculations, the results may not be what we expect. For instance, in calculating interest earnings on a savings account over a number of years, the interest earned must be calculated, say, every quarter and the interest added to the current principle before making the next interest calculation. At each step, the result should be rounded. If it is not rounded, subsequent calculations will be based on numbers that may be slightly incorrect. In addition, to format numbers on the Commodore 64 and VIC 20 computers the number must first be rounded.

All Math Method

The technique for rounding that relies completely on mathematics depends on the INT (integer) function for eliminating digits that are to the right of the point at which we wish to round. The INT function eliminates all digits to the right of the decimal point and returns the next lower whole number. In practice, the INT of 1.1 is 1, of 1.9 is 1, of −1.1 is −2 and of −1.9 is −2. As you can see, INT acts in a somewhat unexpected manner when handling negative numbers. Also, it does *not* round before truncating. But ours is not to reason why, ours is to understand *how* it works and then program it to get the results we need. Key in and RUN the following program to experiment with the INT function.

```
10 INPUT N
20 PRINT INT(N)
30 GOTO 10
```

To observe INT at work, the number entered must have at least one number to the right of a decimal point. Try 1.1, 1.9, −1.1 and −1.9 to verify the examples given above. Press the RUN/STOP and RESTORE keys in combination to exit the program.

Fine, now that we know how INT works, how do we use it to round? The trick is to add .5 to the number before using the INT function. Change line 20 to read

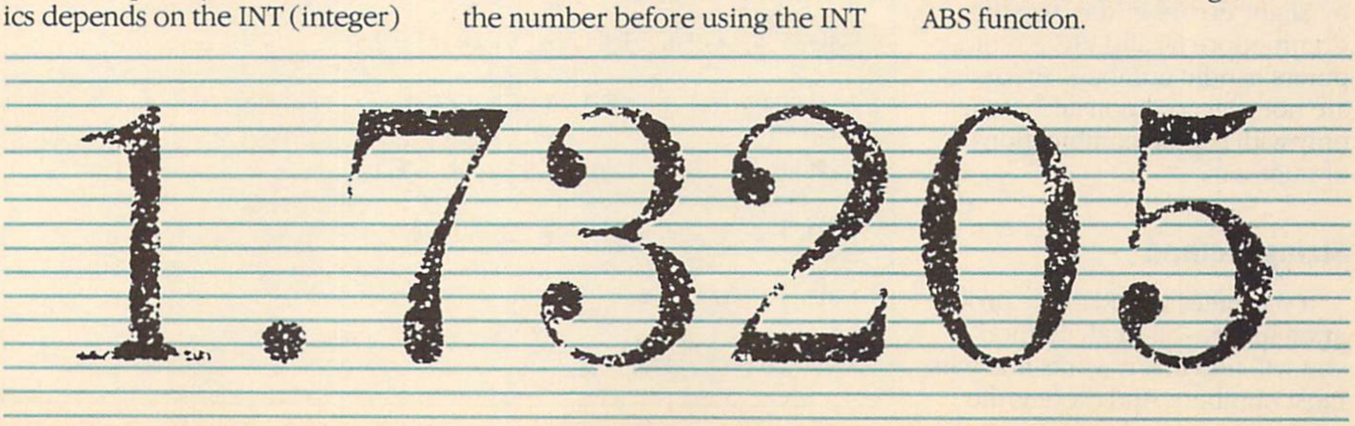
```
20 PRINT INT (N + .5)
```

and try the program again. Works fine with positive numbers, but what about negative numbers? (Try −1.5, which should round to −2.)

To be able to round negative numbers as well as positive numbers using INT, we must elicit the aid of two other mathematical functions: ABS and SGN. Change line 20 to read,

```
20 PRINT SGN(N)
    * INT (ABS(N) + .5)
```

and try the program again. Works fine now! The ABS function returns the absolute value of N. If the value of N is positive, there is no change, but if N is negative, then ABS returns a positive N. This allows the formula for rounding to work correctly (we saw above that positive numbers round correctly). The SGN is used to restore negativity. If N is positive, SGN(N) is one and multiplying by one will have no effect on the rounded value. But if N is negative, SGN(N) is −1 and multiplying by −1 will restore the rounded value to its original negative condition, a condition that was lost using the ABS function.



We can now round any value, positive or negative, to a whole number, but what do we do to round to one, two or more places to the right of the decimal point? After all, much of our data processing will deal with money and we will usually need to round to two places to the right of the decimal point. Actually, it is pretty easy.

To round to two places to the right of the decimal point we will first multiply by 100 to move the decimal point two places to the right of its present position, then we can round this value, and, finally, we will divide by 100 to move the decimal point back to its original position. Change line 20 to read,

20 PRINT SGN(N) * INT (ABS(N) * 100 + .5) / 100

and try the program again. To see the formula work, you must enter numbers with at least three digits to the right of the decimal point.

The number of digits that remain to the right of the decimal point is controlled by the number you use to multiply and divide. As seen above, multiplying and dividing by 100 leaves two digits to the right of the decimal point. Multiplying and dividing by ten would leave one digit, by 1000 would leave three, and so on.

The formula looks pretty complicated, but I hope that taking its development step by step has enabled you to understand how it works.

If you experiment enough, you will find that this formula will not work in *all* cases. This is not the fault of the formula, but is caused by slight errors in the way the Commodore 64 and VIC 20 computers handle numbers. Errors are not very common and occur only with certain combinations of numbers.

String Method

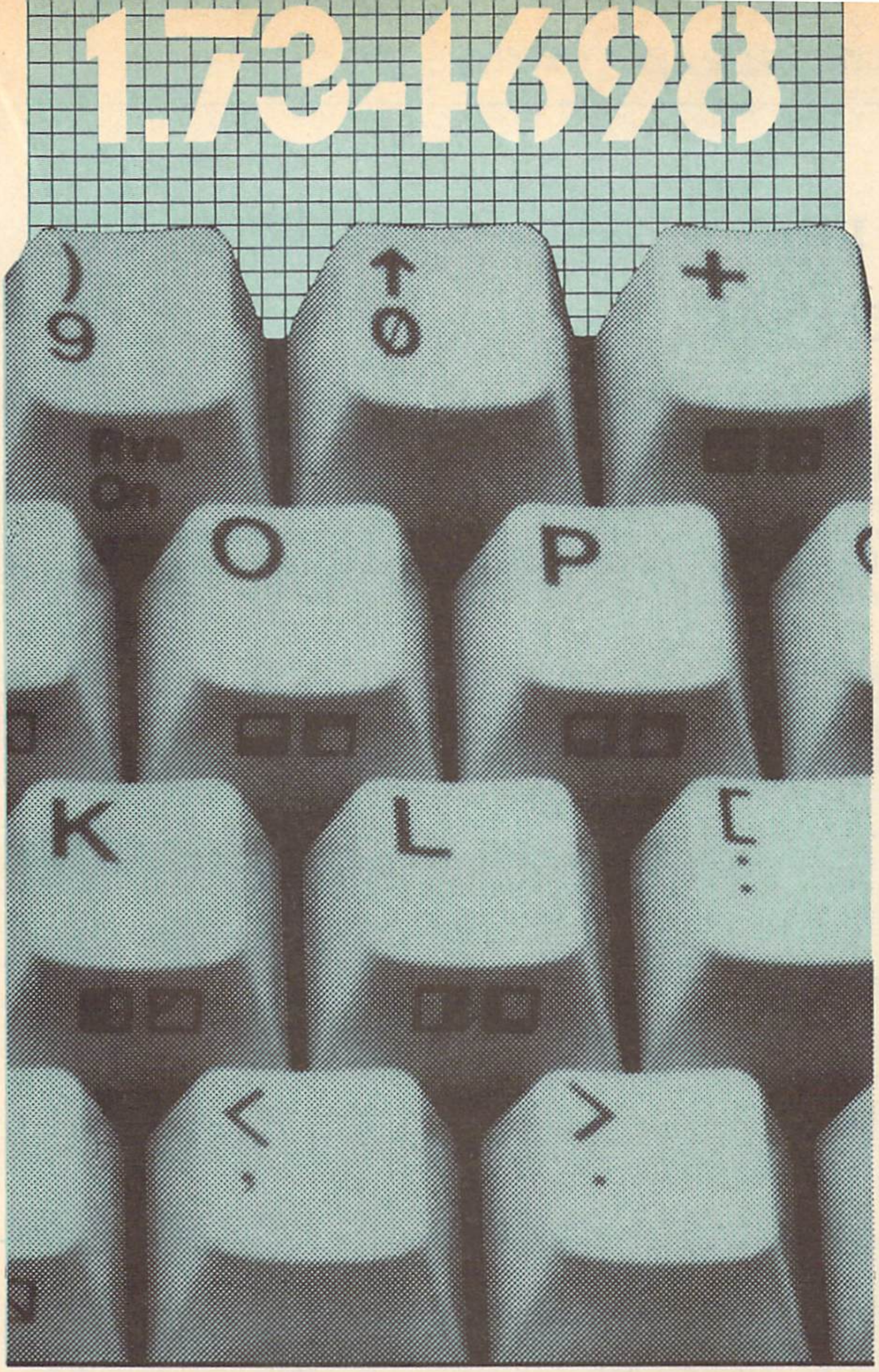
If you enter and RUN the program for the "all-math method" you will find that it works fine with most numbers. And there is the catch—*most!* For example, if you enter 99.335 you would expect to get an answer of 99.34, right? But both the VIC 20 and the Commodore 64 computers return an answer of 99.33. An input of 33333.445 returns 33333.44 and 567.555 return 567.55. The problem seems to exist only when the third number to the right of the decimal point is a five, and then depends on what combination of numbers is to the left of the five. You may want to key in the program and experiment with it a bit. Are there any other conditions under which the problem exists?

While it is true that the rounding problem occurs only under special circumstances, we would prefer that our computer always did its homework "right", right? This problem can be resolved by first converting the number into its equivalent string (STR$), and then using the powerful string-handling functions available to us on the Commodore computers to help us do the rounding.

The following program demonstrates a solution to the problem using strings: The program rounds to two places to the right of the decimal point. What we will be doing is exactly what we do when we round by hand. To round to two places to the right of the decimal point, we look at the third digit to the right of the decimal point. If the number is five or more then we add one to the second number to the right of the decimal point. Finally, we truncate the number to two digits to the right of the decimal point. The order of actions in the program is different, but the principle is the same. You can key it in and try it out, then I will explain its workings (if you haven't figured them out yourself).

And now for a blow-by-blow ac-

```
10 INPUT N
20 F$ = STR$(N)
30 L = LEN(F$)
40 PD = 0
50 FOR I = 1 TO L
60 IF MID$(F$,I,1) = "." THEN PD = I
70 NEXT I
80 IF L <= (PD + 2) THEN 140
90 R$ = MID$(F$,(PD + 3),1)
100 F$ = LEFT$(F$,(PD + 2))
110 IF R$ < "5" THEN 140
120 F = ABS(VAL(F$)) + .01
130 F$ = STR$(F * SGN(N))
140 PRINT F$
150 GOTO 10
```



count of how the program works.
10 Inputs a number from the keyboard and stores it in N.
20 Converts the number stored in N to its string equivalent using the STR$ function and stores the string in the variable F$.
30 Determines the number of characters in F$ (LEN) and stores that value in the variable L.
40 Assigns the value of 0 to the variable PD (for Position of the Decimal point).
50-70 A FOR/NEXT loop that searches for the decimal point. It uses the MID$ function to examine each character in F$ and sets PD equal to the numeric position of the decimal point in the event that the decimal point is found. If the decimal point is not found (a whole number) then the variable PD remains equal to 0.
80 This line checks to see if the number needs to be rounded. If the length of F$ (L) is less than or equal to the numeric position of the decimal point (PD) plus two, then there are no more than two digits to the right of the decimal point and nothing to be rounded. If true, then control passes to line 140 where F$ is printed on the screen.
90 The third digit to the right of the decimal point (PD + 3) is read and saved in the variable R$. This is the value that will determine how the number will be rounded.
100 This line removes the digits to the right of the point at which we want to round. It uses the LEFT$ function to read F$ up to the second digit and replaces the old value of F$ with this new, truncated value.
110 If the third digit (R$) is less than five, then nothing more needs to be done to the value in F$ and program control passes to line 140 to print F$ on the screen.
120 Having failed the test in line 110 means that we need to add a one to the final digit of the number to round up. To accomplish this, we must convert the numeric string in F$ to a numeric value (VAL), take its absolute value (ABS) to avoid problems with negative numbers, and add .01. The result is stored in the variable F.
130 The value of F is now multiplied by the result of SGN(N) which will restore a negative condition if it existed. The result is converted into its string representation (STR$) and the result is stored in F$.
140 The value of F$ is printed on the screen.
150 Sends control to line 10 to ask for another number.

Well, there it is, a solution to the rounding problem. The routine is written to round to two places to the right of the decimal point; it is a relatively easy matter to change it to round to any desired number of digits. A handy addition would be a few lines to format the number so that it always prints two digits to the right of the decimal point, adding a ".00" or "0" as required; these lines would begin at line 140. I will leave this task to you. You may refer to previous issues of *Commodore* magazine for articles that demonstrate the techniques available.

Happy rounding!!! **C**

Solving Relative File Problems on the 1541

by Larry Greenley
Commodore Software

Often users have problems manipulating relative files, especially when it comes to reading a relative file that has already been created on the 1541 disk drive. This problem, however, usually lies in the way the records were *written to* disk rather than the way they are *read from* disk. If you use the following algorithm in a relative file write routine, you will ensure that the data is properly written to disk, which should then eliminate most read errors.

Relative File Write Routine Algorithm

```
10 Open 1,8,15:Rem  "Open the command channel".
20 Open 2,8,2,"file,L,"+chr$(100):Rem  "Open and create the relative file where file=relative file name, chr$(100)=record length".
25 I=1:Rem  "Initialize i (record number) to 1".
30 Input"Enter a record";a$:Rem "Input data to be written to file".
40 Print#1,"p"chr$(2)chr$(I)chr$(0)chr$(1):Rem  "Position the record pointer to the Ith record".
50 Input#1,a,b$,c,d:Rem  "Input error channel variables".
60 If a < 20 then 100 :Rem  "Check for errors less than error 20,if the error is less than 20 ignore it and go to line 100".
70 If a=50 then print#2,0:go to 40: Rem  "Check for error 50 (record not present)".
74 Rem  "If it exists write (print#2) a dummy record (0) to the disk in order to make a record present and eliminate error 50".
76 Rem  "Go back to the original position command (line 40) and re-position the record pointer to the same place".
80 Print a,b$,c,d:Rem  "If any other error occurs,print it and stop".
100 Print#2,a$:Rem  "This statement writes the data to the file on disk".
```

```
105 Input"Do you want more records(y/n) ";g$:if g$="n"then 120:Rem
"Checks to see if more records are needed,if not, go to line 120".
110 I=I+1 : Go to 30:Rem  "Increment record number (I) then go to
line 30 to input another record".
120 close 1:close 2:end:Rem    "Close both previously opened channels
and end".
```

During a relative file write routine, it is absolutely necessary to check for error number 50—"record not present". This is an error condition that can be "trapped" (checked for and corrected at run time) by inputting the error channel variables as in line 50, checked as in lines 60 and corrected as in line 70. Line 60 is the crucial statement that ensures that the data is written to the disk correctly. If error 50 is not checked and it does occur when writing to a relative file, the record pointer defaults to position one and destroys the original record, usually replacing it with a null record or bad data ( ). Then the disk drive either spins continually with a red flashing light or it locks up with a solid red light. In either case, read the error channel to clear the error and reset the drive.

If error 50 is detected as in line 70, a dummy (filler) record is written to the file and control is returned to the original position command (line 30) for re-positioning. The dummy record (0) creates a record that had not previously been present. Once the record pointer is repositioned and no other errors are detected, the data record can then be written to the file. C

Nevada COBOL™ and Nevada FORTRAN™

by April M. Koppenhaver

Nevada COBOL and Nevada FORTRAN are Commodore's versions of the ANSI standard COBOL and FORTRAN languages. Each program has been designed for small businesses and uses the Commodore 64 CP/M® operating system cartridge.

A data disk is included in both the Nevada COBOL and Nevada FORTRAN software packages. To prevent mistakes from occurring, these disks should only be read. Therefore, both the Nevada COBOL and Nevada FORTRAN manuals include directions on how to create a CP/M operations disk using your CP/M operating system disk. You can then save, compile, and modify your Nevada COBOL and Nevada FORTRAN programs on their respective operations disks.

When building a program, your first step is to create source code. You can use the text editor ED.COM to help you. (ED.COM is one of the files to be copied onto your operations disk.) Next, compile the source code. If any compiling errors are found, the particular line number of the program and an error message are displayed. If errors do appear, correct the source program and recompile the code. When a program has been successfully compiled, an object file is created. Once the object file has been produced, the program can be executed by simply typing RUN and the file name.

Both the Nevada COBOL and Nevada FORTRAN manuals are written for the experienced programmer. It is also assumed that you have read the Commodore 64 CP/M operating system user's guide.

Nevada COBOL

Nevada COBOL for the Commodore 64 is an updated version of ANSI-1974 COBOL. COBOL is an acronym for Common Business Oriented Language. It has been used since the 60's to meet the needs of many business applications.

COBOL is based on English and uses certain words and syntax rules derived from English. As in English, the basic unit of COBOL is a word. A "word" may be a COBOL reserved word or a word that you define. Reserved words have a built-in, specific meaning to the COBOL compiler. Programmer-defined words must conform to the COBOL rules for name formation and can be assigned to data names and procedure names.

As the programmer, you can combine reserved words and your programmer-defined words into clauses and statements. A clause or a statement specifies one action to be performed, one condition to be analyzed or one description of data. These clauses and statements can then be combined into sentences.

Sentences may be simple (one statement or clause) or they may be compound (several statements or several clauses). Logically related sentences can be combined into paragraphs. Related paragraphs can be combined into sections. These statements are then placed in one of the appropriate program divisions. Here are the four divisions of a COBOL program:

Identification Division
Environment Division
Data Division
Procedure Division

The Identification Division is entirely for documentation purposes and is treated as comments by the compiler. This is where you specify the program name, programmer's name, application, dates when written and compiled, and any security restrictions. Here is an example of an Identification Division.

```
0001   Identification Division.
0002   Program-ID
0003          T6WF.
0004*  This program creates a file or fixed length
0004*  records if the record sizes are changed to
0004*  your needs, can be used to create the space
0004*  needed (allocate) for a random file.
```

The Environment Division is used to specify the name of the source computer (used to compile the program) and the name of the object computer

Nevada COBOL, Nevada FORTRAN and Nevada ASSEMBLER are trademarks of Ellis Computing Inc. CP/M is a registered trademark of Digital Research, Inc.

(used to execute the compiled object program). The Environment Division may consist of a Configuration Section, Input/Output Section for identifying the name of each file used and the associated external hardware devices. Here's an example of an Environment Division.

```
0005   Environment Division.
0006   Configuration Section.
0007   Source-Computer.
0008         8080-CPU.
0009   Object-Computer.
0010         8080-CPU.
0011   Input-Output Section.
0012   File-Control.
0013         Select file name up to 30 characters
0013         Assign to disk hardward I/O device
             program
0013         is sent to
0014            Organization is sequential.
0015            Access mode is sequential.
```

The Data Division is for specifying a detailed description of all the data to be used. This includes the format of each file and records within the files. In the Data Division, you assign data names to each of the data items to be used. Also in this division is the Working Storage Section. This is where you describe records and data items that are not part of the files, but are used during the processing of the object program.

```
0016   Data Division.
0017   File Section.
0018   FD FILE1
0019         Label records are standard
0020         Value of File-ID is out-file-name
0021         Block contains 1 record
0022         Data record are O-Record
0023   01 O-Record
0024         02 SEQ PIC 9999.
0025         02 REC1 PIC is X(156).
0026         O2 SEQ2 PIC 9999.
0027   Working-Storage Section. Contains
0027   Intermediate Storage Area along with
       constant
0027   Values used
0028   01 OUT-OF-FILE-NAME PIC X(14)
0029            Value "A:TESTF. WRK".
0030   01 X1 PIC 9999
0031            Value 0001.
```

The Procedure Division is for defining instructions for solving the given problem. Here is an example of a Procedure Division:

```
0032   Procedure Division.
0033   Begin.
0034         Display "Enter output file name".
0035         Display out-file-name with no advancing.
0036*  to accept and use the file-name just displayed
       you can hit
0036*  the   CR   key. see #2 under accept.
0036         Accept out-file-name.
0037         Open output FILE1.
0038         Move spaces to O-Record.
0039   BEGIN2.
0040         Move X1 to SEQ.
0041         Move X1 to SEQ2.
0042         Add 1 to X1.
0043         Display O-Record.
0044         Write O-Record.
0045         If X1 is = to 201
0046               Go to EOJ.
0047         Go to BEGIN2.
0048   EOJ.
0049      Close FILE1.
0050      Stop run.
0051   End program T6WF.
```

The Nevada COBOL reference manual includes a general discussion of certain COBOL programming concepts, a list of COBOL reserved words, a detailed description of the four divisions of a COBOL program, several sample programs, a list of error codes and messages and a glossary of terms.

Nevada FORTRAN

Nevada FORTRAN for the Commodore 64 is an updated version of ANSI Standard FORTRAN. FORTRAN is a widely used programming language for scientific research and developmental applications.

The FORTRAN character set consists of the entire alphabet, numbers zero through nine, and special characters. The special characters include the blank, equal sign, plus sign, minus sign, asterisk, slash, left and right parenthesis, comma, decimal point, dollar sign, number sign, ampersand and backslash. The Nevada FORTRAN manual details the meanings of these special characters as they are used in this version of FORTRAN.

Nevada FORTRAN includes numerous extensions to the version X3.9-1966 of ANSI Standard FORTRAN. Here is a list of the features:

- Free-format input and output
- IMPLICIT statement for setting default variable types
- Options end-of-file and error branches in READ and WRITE statements
- COPY statement to insert source files into a FORTRAN program
- Direct inline assembly language
- Access to file system for such functions as creating, deleting, and renaming files
- Random access on a byte level to files
- Access to absolute memory locations
- Program controlled time delay

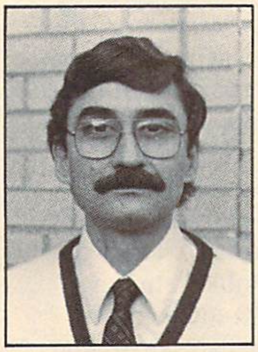
- A pseudo random number generator function
- Program control of runtime error trapping
- Ability to chain a series of programs
- Ability to download object code into memory
- CALL function to execute previously loaded code
- Program tracing
- IF-THEN-ELSE statement
- Enabling and disabling console abort of program
- ENCODE and DECODE memory to memory I/O
- Multiple returns from subroutines
- K format specification

The Nevada FORTRAN manual includes some general concepts and details of FORTRAN programming, summaries of system functions and subroutines and a list of run-time and compile-time errors. The Nevada ASSEMBLER™ software and manual are also included in the Nevada FORTRAN package. **C**

On Polynomials

Part 1, The Cubic Equation

by Shlomo Ginsburg
University of Kansas, School of Engineering



Dr. Shlomo Ginsburg

This article discusses mathematical entities called polynomials. I have assumed that readers have some familiarity with algebra—for example that they know how to solve a quadratic equation. The interested novice should refer to an elementary textbook and to the references provided here.

A polynomial of degree n is an expression of the form

$$P = p_0 + p_1 X + p_2 X^2 + p_3 X^3 +, \ldots, + p_n X^n = \sum_{k=0}^{n} p_k X^k \tag{1}$$

where the coefficients p_k $(k=0,1,2,\ldots,n)$ are assumed here to be real numbers. The notation on the righthand side of equation (1) is called the "summation" symbol. When the polynomial P is equal to zero we have a polynomial equation. Here, we shall discuss the following equation

$$P = p_0 + p_1 X + p_2 X^2 + p_3 X^3 = 0 \tag{2}$$

which is a *cubic equation.*

Dividing equation (2) by $p_3 \neq 0$ (if $p = 0$ we have a quadratic equation, so we assume that $p_3 \neq 0$) we get

$$a_0 + a_1 X + a_2 X^2 + X^3 = 0 \tag{3}$$

where $a_0 = p_0/p_3$, $a_1 = p_1/p_3$, $a_2 = p_2/p_3$.

We now define the following relations

$$q = (3a_1 - a_2^2)/9 \tag{4}$$

$$r = (9a_1a_2 - 27a_0 - 2a_2^3)/54 \tag{5}$$

$$s = \sqrt[3]{r + \sqrt{q^3 + r^2}} \tag{6}$$

$$t = \sqrt[3]{r - \sqrt{q^3 + r^2}} \tag{7}$$

$$d = q^3 + r^2 \tag{8}$$

Using these definitions we can express the solution as follows:

Case 1: $d \geq 0$ (d is greater than or equal to 0)

$$x = s + t - a_2/3 \tag{9}$$

$$x = -(s+t)/2 - a_2/3 + i\sqrt{3}\,(s-t)/2 \tag{10}$$

$$x = -(s+t)/2 - a_2/3 - i\sqrt{3}\,(s-t)/2 \tag{11}$$

where i denotes the complex number $\sqrt{-1}$. Note that when d equals zero we get $s=t$, and therefore we have three real roots, two of which are identical. Otherwise, we have one real and two complex roots.

Case 2: $d < 0$ (which implies $q < 0$, as r is non-negative). For this case there are three distinct real roots. With the definition

$$\cos\theta = r/\sqrt{-q^3} \tag{12}$$

we get

$$x_1 = 2\sqrt{-q}\,\cos(\theta/3) - a_2/3 \tag{13}$$

$$x_2 = 2\sqrt{-q}\,\cos(\theta/3 + 2\pi/3) - a_2/3 \tag{14}$$

$$x_3 = 2\sqrt{-q}\,\cos(\theta/3 + 4\pi/3) - a_2/3 \tag{15}$$

Note that the angles are measured in radians ($2\pi/3$ equals 120°, etc.).

Apparently, the calculation of the roots is quite simple. As an example, we may take the case $(x-1)(x-3)(x+6) = 18-21x+2x^2+x^3$, with the solutions $1,3,-6$. When we solve this case with a calculator, the angle is found by computing $\cos^{-1}(r/\sqrt{-q^3})$ and then the expressions (13)−(15). But, when we use a computer, which does not provide for the inverse function $\cos^{-1}$, it is necessary to use the (trigonometric) relation

$$\tan^{-1}\theta = \sqrt{1-\cos^2\theta}/\cos\theta \tag{16}$$

and find θ from the inverse function $\tan^{-1}\theta$. This will cause some problems if we are not aware of

the limitations of this computation, especially the range of definition of the function involved. To see this, try the same example as before using $\theta=\tan^{-1}(\sqrt{-d/r})$ which is the correct expression derived from (12) (try to get it yourself). You do not get the same answer!

Instead of discussing the calculations on a theoretical basis, we shall use a program for demonstration. Before writing a computer program it is advisable to draw a flow chart. The drawing is essential for avoiding errors and is used by experts, not just beginners as one may think. For the cubic equation we use the simple flow chart shown in Figure 1.

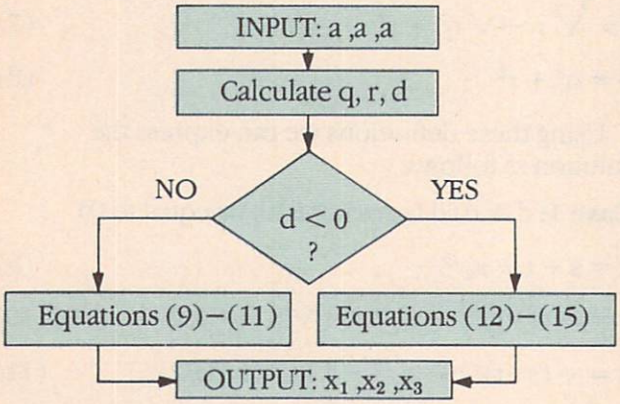


Figure 1. Flow Chart for Cubic Equation Program

This flow chart is very general, and requires some elaboration within each block. For example, the computations of equations (12)−(15) will need another flow chart, which explains how to proceed when r=0, as one cannot compute $\tan^{-1}\theta$ for this case. To clarify matters we shall examine the computer program now, and then summarize the lesson.

It is assumed that the reader is familiar with Commodore BASIC. The program will run on the Commodore 64 as is, and on most computers after deleting machine-dependent commands. Here, the latter consist of color selections, screen-clears, etc. The main parts of the program are:

1. Calculation of Q,R,D in lines 220-240,

2. Calculation of equations (6), (7), (9)−(11) in lines 260-330,

3. Calculation of the angle θ and equations (13)−(15) in lines 400-460.

There are several changes in the expressions used within the program, compared with the mathematical derivation. These are of major importance and will be discussed in detail.

First, we note that instead of using A2↑2 in line 220, we use A2*A2. This is a more accurate calculation. We also compute S and T only if D>0 (lines 270-300) to save time and memory. We use some peculiar expressions on lines 290-300 for S and T. The reason is that we need the cubic root of (R+D) and (R−D). These roots can be obtained regardless of the sign in direct mode (try −8↑(1/3) to get the correct result −2). But, the Commodore 64 cannot compute A↑(1/3) in a program if A<0. Using the SGN function (sign) is a fast way to obtain the correct result; it is a modification of Commodore's built-in function. We shall see later that we need modify another function, too. The last peculiarity in this section is the computations in lines 320-330. Obviously, the values of X2 and X3 are not the roots we are looking for, but some intermediate calculations. We use these for two reasons: the Commodore 64 cannot handle the complex numbers for us, so we need to make the distinction between real and imaginary parts; and we save memory by using three variables (X1,X2,X3) only. We used also the variable D in line 260 to store the value of the square root of D, thus saving memory. This is good practice, but one has to be careful not to change a value which may be required later.

In the case of three real roots, lines 400-460, we find additional funny expressions which need to be explained. We begin with the arbitrary value $T=\pi/6$. This is the correct value for R=0 (see equation (12)). If R=0 the computer, unlike a human, will try to divide by zero in line 410 and stop running with an ERROR message. One can skip to line 430 whenever R=0. Otherwise the two expressions in lines 410 and 420 are computed according to the sign of R. This is probably the most difficult part of the algorithm. The distinction between R>0 and R<0 is a result of the specific ATN function ($\tan^{21}$) used. It assumes angles between $-\pi/2$ to $+\pi/2$. If R is negative the angle cannot possibly be in this range (see equation (12), remember that here Q<0). Thus, we need to add π degrees to the value obtained in line 410 to account for negative cosines. We do that in line 420.

We have learned about the solution of cubic equations, with and without computers. The algorithm for finding the roots of the third degree polynomial is quite simple. As an exercise, it is suggested that the reader write a program for solving the roots of a quadratic equation.

In future articles more complicated equations, such as polynomials of higher degree and other functions, will be discussed. Generally, these cases require a totally different approach, since closed-form solutions are not available. We shall learn about numerical methods, and see that the computer is a powerful tool for these cases.

Regarding the use of computers for the cubic equation and for other applications, we learned the following lessons:

1. Define the problem and summarize the solution in a flow chart.

2. Check all expressions and functions used within your program to avoid errors. Typical mistakes lead to division by zero and using built-in functions beyond the range of definition (the computer's, not the theoretical range).

3. Use test cases to check the solution.

4. Try to avoid unnecessary computations, and to save memory. You may use the same variables to store different values (D is used for D or for SQR(D) in our case). But remember that once you change the value of a variable the previous one is lost. **C**

References

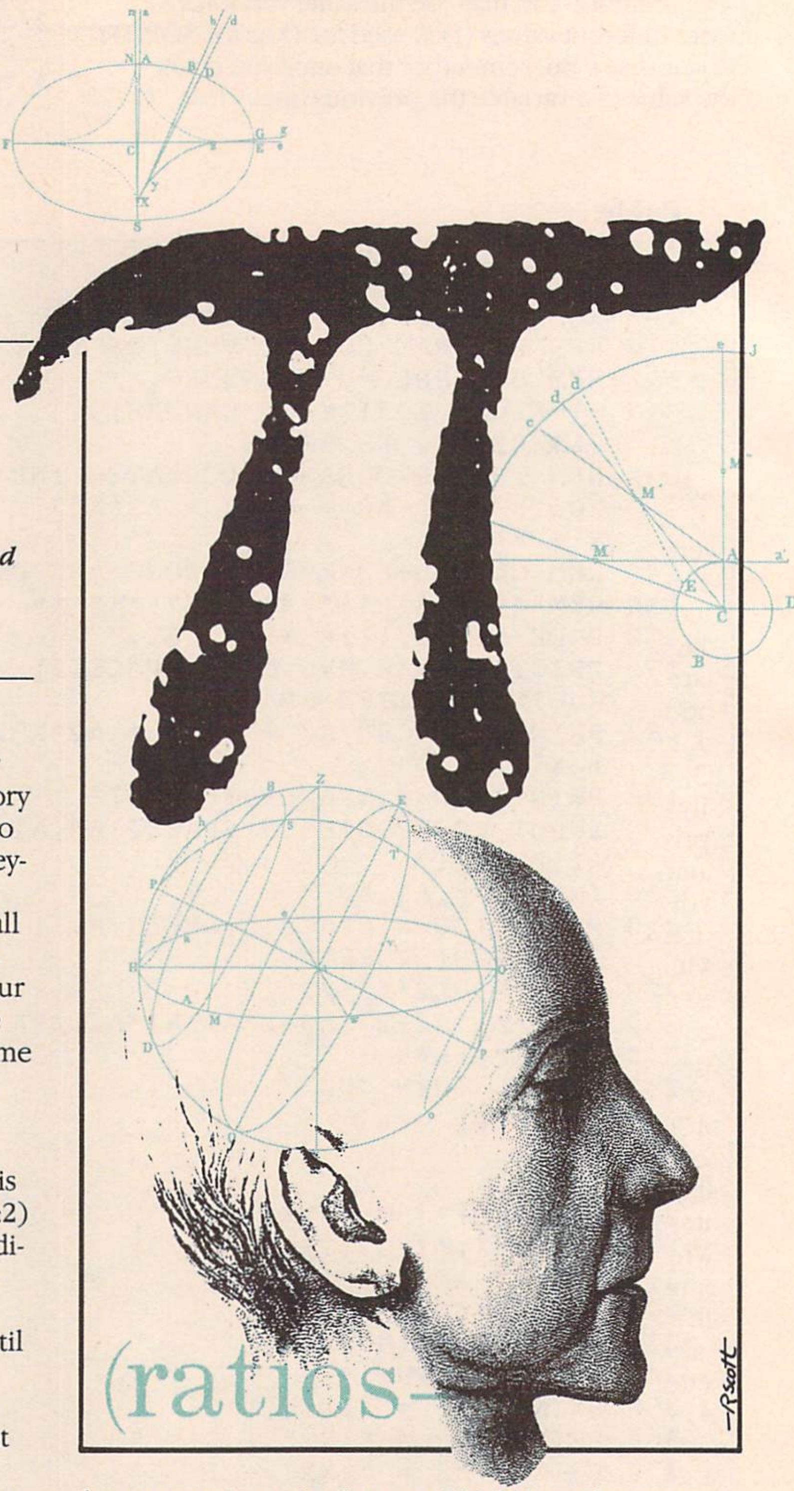
1. Curchill, R.V., *Complex Variables and Applications,* McGraw-Hill, 1960.
2. *Commodore 64 Programmer's Reference Guide,* Commodore Business Machines, Inc., 1983.
3. Spiegel, M.R., *Mathematical Handbook,* McGraw-Hill, 1968.

Cubic

```
100 REM ***************************
110 REM PROGRAM CUBIC, VERSION I, 1984
120 REM BY: SHLOMO GINSBURG,
     THE UNIVERSITY OF KANSAS,
     LAWRENCE, KS 66045
130 REM THIS PROGRAM CALCULATES THE
    ROOTS OF:  A0 + A1*X + A2*X^2 +
    X^3 = 0
140 REM CLOSED-FORM SOLUTIONS ARE USED
150 REM ***************************
160 POKE 53280,11:POKE 53281,0
170 PRINT"[CLEAR,RVS,BLUE,SPACE13]
    CUBIC EQUATION[SPACE13]"
180 PRINT"[SPACE6]A0 + A1*X + A2*X^2
    + X^3 = 0"
190 PRINT"[DOWN,SPACE6,YELLOW]
    INPUT COEFFICIENTS[SPACE2]A0,A1,
    A2[DOWN2]"
200 INPUT A0,A1,A2
210 PRINT"[DOWN,SPACE15,GREEN,RVS]
    SOLUTION[RVOFF]"
220 Q=(3*A1-A2*A2)/9
230 R=(9*A1*A2-27*A0-2*A2*A2*A2)/54
240 D=Q*Q*Q+R*R
250 IF D<0 THEN 400
260 D=SQR(D)
270 S=R+D
280 T=R-D
290 S=SGN(S)*(SGN(S)*S)^(1/3)
300 T=SGN(T)*(SGN(T)*T)^(1/3)
310 X1=S+T-A2/3
320 X2=-(S+T)/2-A2/3
330 X3=SQR(3)*(S-T)/2
340 REM * OUTPUT *
350 PRINT" X1 = ";X1
360 PRINT" X2 = ";X2" + I* "X3
370 PRINT" X3 = ";X2" - I* "X3
380 GOTO 500
390 REM * ALL ROOTS REAL AND DISTINCT
400 T=[PI]/6
410 IF R<>0 THEN T=ATN(SQR(-D)/R)/3
420 IF R<0 THEN T=T+[PI]
430 Q=2*SQR(-Q)
440 X1=Q*COS(T)-A2/3
450 X2=Q*COS(T+2*[PI]/
    3)-A2/3
460 X3=Q*COS(T+4*[PI]/
    3)-A2/3
470 PRINT" X1 = ";X1
480 PRINT" X2 = ";X2
490 PRINT" X3 = ";X3
500 PRINT"[DOWN,YELLOW,
    SPACE12] NEW CASE
    ([BLUE]Y[YELLOW]/
    [RED]N[YELLOW])"
510 GET A$:IF A$=""
    THEN 510
520 IF A$<>"N" THEN 170
530 PRINT"[DOWN3,RVS,
    BLUE,SPACE15]
    GOOD BYE[SPACE17]"
540 END
```

Calculating Pi

by Craig R. Hessel



Ever get to wondering about the number π? Here are some facts you may not have known and a LIAL program that can compute π accurately to several hundred decimal places on any Commodore computer. LIAL (Large Integer Arithmetic Language) is described in "Number Crunching on the 6502" (Issue 28) and an interpreter for the language is contained in "Public-Key Cryptography for Commodore Microcomputers" (Issue 26).

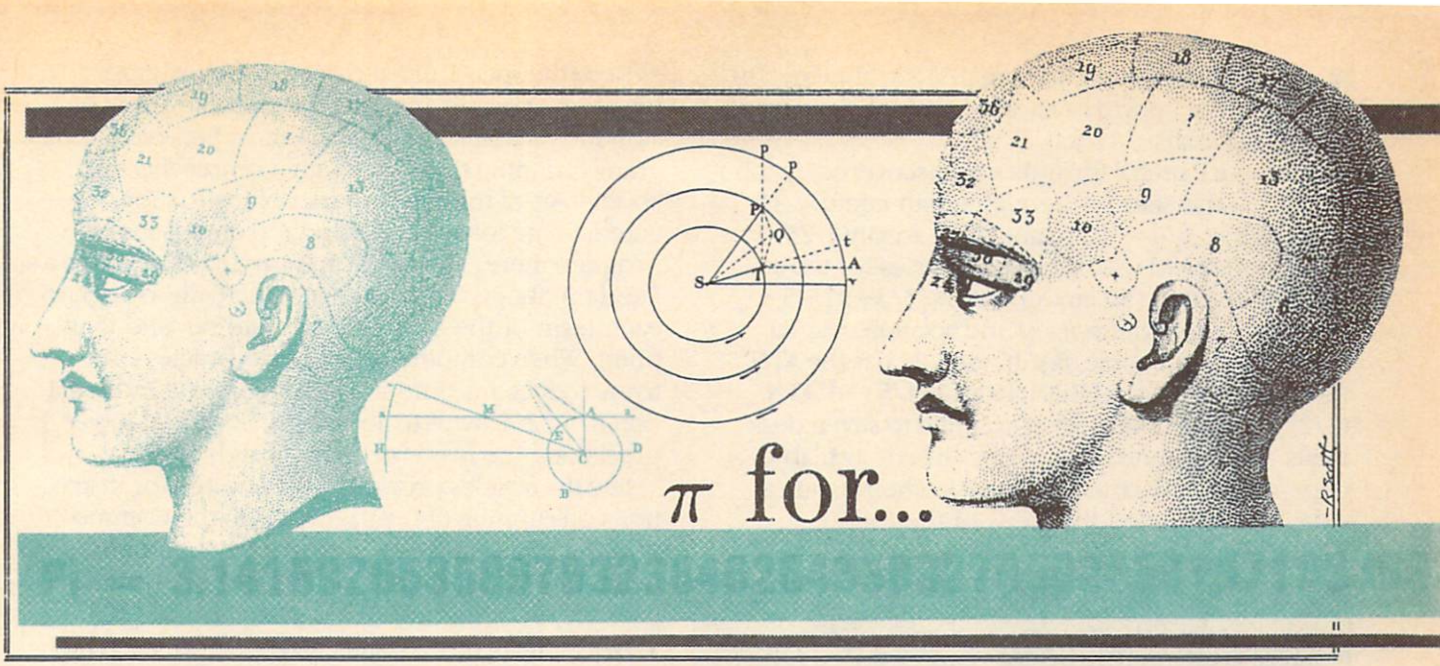
The number π represents the ratio of the circumference (perimeter) of a circle to its diameter (width). π occupies a prominent place in the history and study of numbers. The fact that π is assigned to one of the keys on your Commodore computer keyboard is an indication of its importance.

What is the number value for π? You might recall from school that 22/7 has something to do with π. Or if you PRINT π, then 3.14159265 appears on your screen. Both of these numbers, it turns out, are close to but not equal to π. To see where these came from and why we bother with approximations rather than an exact value, let's back up and start from the beginning.

The earliest approximation to π from antiquity is 3. There is even a biblical passage (2 Chronicles 4:2) which describes a circular object with a ten cubit diameter and a 30 cubit circumference. The ancient Egyptians used (4/3)↑4 for the ratio, although the origin of that expression is not clear. It was not until Greek mathematics flourished that the search for the value of π became more than a matter of measuring or empirical guesswork. Euclid was the first to show that the ratio of the circumference of a circle to its diameter is the same for all circles, large and small. Then Archimedes, about 2200 years ago, proved that π is between 223/71 and 22/7. He repeatedly "sandwiched" the circle between pairs of regular polygons (many-sided figures with sides of equal length), whose perimeters he could calculate. At each step, by using polygons with twice as many sides as before, he was able to get a better estimate for π. Archimedes stopped with 96-sided polygons.

In fifth century China, Tsu Chung-Chih used similar geometric methods to get 355/113 as a better approximation yet to π. After about 1000 years of little progress, the Persian mathematician al-Kashi calculated π to 16 decimal places. This approximation stood for about two centuries. Then, in the 15 years before his death in 1610, Ludolph van Ceulen laboriously calculated π to 35 decimal places. As a result of his effort, π is sometimes (at least in Germany) still referred to as Ludolph's number.

In succeeding years, better ways for computing



π were discovered, and various individuals were able to improve the estimate for π. The chronology includes John Machin, 1706, 100 decimal places; Zacharias Dase (who once mentally calculated the square root of a hundred digit number in less than an hour), 1844, 200 decimal places; William Rutherford, 1853, 400 decimal places; and William Shanks, 1873, 707 decimal places. These were all extraordinarily time consuming calculations by hand and it is surprising that most were error free. But, alas, to err is human, as the bard once noted. In 1946, a mistake was discovered in the Shanks approximation, ruining it from the 528th decimal on.

With the advent of the computer age, progress has been made by leaps and bounds. After all, repetitive calculation is a job that computers excel at. In 1949, π was computed to 2037 decimal places in 70 hours (!) on the ENIAC computer. A dozen years later, an IBM 7090 churned out π to 100,000 decimal places. Since then the result has surely been extended to 1,000,000 decimal places or more at some computer center somewhere.

Why all the fuss? It's unlikely that you or I would ever need a better approximation to π than Chung-Chih's 355/113, which is accurate to six decimals. It's also unlikely that any scientist would ever need a better estimate than van Ceulen's: 3.14159265358979323846264338327950288.

Up until the mid-eighteenth century, though, mathematicians still had a good reason for the lengthy computation of π. There was the hope of finding a repeating pattern in its digits. Then π could be written explicitly, once and for all, as the ratio of two (probably very large) integers. But this hope was dashed in 1761 when J. H. Lambert proved that π, like the square root of two, is an irrational number. This meant that π could never be represented exactly as a fraction or written in a precise decimal form. Still, the calculating went on.

With the development of computers as a calculating tools, there has been some consideration given to the use of the decimal expansion of π as a source of random digits. But that may be more a rationalization than a reason for the effort. People compute π for the same reason they sit on flagpoles or swallow goldfish. The challenge is always there to do the record one better.

There are many ways to estimate π—some useful and some not so useful. There are even probabilistic methods. In 1760, Comte de Buffon discovered that π could be estimated with a needle and a sheet of paper. First fill the sheet, in one direction only, with lines evenly spaced the needle's length apart. Then randomly drop the needle onto the paper and check whether or not the needle rests across a line. Do this as many times as patience allows. Call D the number of "drops" and H the number of "hits". Then 2*D/H should approximately equal π. This comes from Buffon's surprising result that the probability of the needle landing on a line (i.e., a "hit") is $2/\pi$.

In the seventeenth century, John Wallis discovered that $\pi/2 = (2/1)^*(2/3)^*(4/3)^*(4/5)^*(6/5)^*(6/7)^*\ldots$, so you might try to find π with this short BASIC program:

```
100 PI=2:N=2
110 PI=PI*N/(N-1)*N/(N+1)
120 PRINT N,PI:N=N+2:GOTO 110
```

When you run the program, the displayed values converge slowly toward π. But when N reaches about a quarter million or so, line 110 no longer changes the variable PI. This is due to the limited

precision of computer floating point arithmetic. By this point, the approximation to π is only accurate to four decimals.

A far more useful formula was discovered by John Machin at the start of the eighteenth century. He showed that π/4 = 4*arctan(1/5)−arctan(1/239). When combined with the power series for the arctangent function (arctan(x) = x−x↑3/3+x↑5/5−x↑7/7+...), this yields a fast and accurate way to compute π. Of course, BASIC already has the ATN function, so you can PRINT 16*ATN(1/5)−4*ATN (1/239) to get a value for π accurate to seven decimals. But this function was not directly available to people like Machin and Shanks. Their hand calculations paralleled the steps of the following BASIC program:

```
100 X=5:GOSUB 200:PI=16*SUM
110 X=239:GOSUB 200:PI=PI-4*SUM
120 PRINT PI:END
200 POWER=X:SUM=0:SIGN=1:N=1
210 TERM=1/(POWER*N)
    :SUM=SUM+SIGN*TERM
220 N=N+2:SIGN=-SIGN
    :POWER=POWER*X*X
230 IF TERM>1E-9 THEN 210
240 RETURN:REM SUM IS ARCTAN OF
    1/X
```

This is a dramatic improvement in speed and accuracy over the Wallis method. Note that the computation of the arctangent function is stopped when the terms in the sum become smaller than .000000001. This reflects the fact that BASIC computes only numbers with nine or so digits of accuracy. With calculations of higher accuracy, more (smaller) terms would be included. But multiplying and dividing with more than nine-digit accuracy is hard to do either in BASIC or by hand. To get around this problem, we'll use LIAL—a high precision language.

The LIAL program listed separately follows the steps of this last BASIC program, but with three notable differences. First, multiplication has been replaced entirely with shifts and additions. A left shift is a fast way to multiply by two. This speeds up the program considerably. Second, a subroutine has been included to convert the binary result to decimal form. Like the ATN function, automatic binary-to-decimal conversion by BASIC is a convenience easily taken for granted. Third, calculations are carried out in a fixed point format rather than the floating point format used by BASIC. Floating point arithmetic is a convenient way to handle numbers in the range 10↑−38 to 10↑38 with only nine- or ten-digit accuracy, which is fine for most applications. For this application, though, we need fixed point precision.

Both the speed and accuracy of the program hinge on line 046. At maximum precision, that division instruction accounts for almost 90% of the program's running time, although it represents only about 0.3% of the total number of instructions executed. Some loss of precision also creeps into the program there. The division is carried out only to a fixed number of decimal places (actually, bytes), so each term of the sum is inaccurate beyond that point. When combined, these inaccuracies effectively reduce maximum precision for the π calculation from 255 bytes to about 253.5 bytes. At lower precisions, the loss is proportionately less.

See the articles previously mentioned for instructions on running LIAL programs. On the Commodore 64, you might allocate the 2.5+ K of memory needed for the program as follows: interpreter package, $C000-$C42D; LIAL program, $C42E-$C4CF; interpreter storage, $C4D0-$C54F; and LIAL storage, $C550-$CA4A. That is a high RAM area protected from BASIC. On the VIC or PET/CBM, you'll need to protect 2.5+ K elsewhere. Before calling the interpreter, remember to set the memory allocation pointers, the table entries for the four subroutine labels, and the initial Z-register value for the precision. After execution stops, the decimal expansion for π begins at the start of LIAL storage, with two digits packed into each byte. This can be readily viewed with a machine language monitor. (Note: INC $DC01, RTS is a suitable "break" detection subroutine for the Commodore 64. INC $9121, RTS works on the VIC. Only the PET/CBM routine was given in the "Number Crunching..." article in Issue 28.)

Run the program first with 16-byte precision. It will take only five seconds to compute π to better than 35 decimal places. That's quite a contrast to the 15 years it took van Ceulen. Next, increase the precision to 43 bytes and run the program again. In just one minute, your computer will surpass the 100 decimal place effort of John Machin, to whom we are indebted for the method of calculation. Then if you have the patience, push the precision to the maximum 255 bytes and call the interpreter once more. In less than three hours, you will have a value for π accurate to 610 decimals. Just 40 years ago that would have been the best approximation known—at least until the 30 tons of tubes and wires called ENIAC tackled the problem. For the record, here are the first and last few digits in the 610-decimal place (611-digit) approximation to π: 3.1415926535...0005681271.

If all this has whet your appetite, you might try reading *A History of Pi,* second edition, Golem Press, 1971, by P. Beckman. Or else see *Asimov on Numbers,* Doubleday, 1977, by I. Asimov.

Craig R. Hessel is a regular contributor to Commodore Microcomputers.

```
;THIS LIAL PROGRAM COMPUTES PI
;BY MACHIN'S METHOD.  INITIAL
;Z VALUE (1-255) SETS NUMBER
;OF BYTES PRECISION.  FINAL A
;VALUE CONCATENATED WITH VØ
;VALUE IS DECIMAL PI (FOUR
;BITS PER DIGIT).  FINAL V1
;VALUE IS BINARY PI/4.  EACH
;FIVE BYTES OF PRECISION IS
;ABOUT TWELVE DECIMAL DIGITS.
;
;    VARIABLES          REGISTERS
; VØ=POWER=DECPI     RØ=FLAG
; V1=SUM             R1=SIGN
; V2=TERM=DECFRC     R2=HCOUNT
; V3=DENOM=BINFRC    R3=LCOUNT
;                    R4=HIGH
;
;    SUBROUTINE LABELS
;  LØ=PIOVR4=START+$ØØØE
;  L1=ARCTAN=START+$ØØ16
;  L2=MULPWR=START+$ØØ5C
;  L3=CONVRT=START+$ØØ78
;
ØØØ  1D      START LDN Z
ØØ1  3Ø            JSR PIOVR4
ØØ2  16            ASL A
ØØ3  16            ASL A
ØØ4  83            STA BINFRC
ØØ5  Ø5 13         LDA #19
ØØ6  33            JSR CONVRT
ØØ7  8Ø            STA DECPI
ØØ8  Ø5 Ø1         LDA #1
ØØ9  33            JSR CONVRT
Ø1Ø  AØ            EXA DECPI
Ø11  1Ø            END
;PIOVR4 SUBROUTINE COMPUTES
;A←SUM←PI/4.
Ø12  12     PIOVR4 ZER A
Ø13  Ø6 ØØ         LDZ #Ø
Ø14  31            JSR ARCTAN
Ø15  16            ASL A
Ø16  16            ASL A
Ø17  Ø6 FF         LDZ #255
;ARCTAN SUBROUTINE COMPUTES
;A←A+ARCTAN(1/5) IF Z=Ø OR
;A←A-ARCTAN(1/239) IF Z=255.
Ø18  81     ARCTAN STA SUM
Ø19  5Ø            STZ FLAG
Ø2Ø  51            STZ SIGN
Ø21  Ø5 Ø1         LDA #1
Ø22  8Ø            STA POWER
Ø23  Ø6 ØØ         LDZ #Ø
Ø24  52            STZ HCOUNT
Ø25  53            STZ LCOUNT
Ø26  Ø7 31         BRN ENTER
Ø27  91       EXIT LDA SUM
Ø28  11            RTS
;MAIN ARCTAN LOOP BEGINS HERE.
;A←POWER*COUNT BY ADDITION
;LOOP SINCE COUNT IS SMALL.
Ø29  1B      OUTLP DEC Z
Ø3Ø  54            STZ HIGH
Ø31  Ø6 ØØ         LDZ #Ø
Ø32  1B       INLP DEC Z
Ø33  FØ            ADD POWER
Ø34  ØD F6         BMI EXIT
Ø35  Ø8 FA    BACK ZNE INLP
Ø36  64            LDZ HIGH
Ø37  Ø8 F3         ZNE OUTLP
;THEN DENOM←2*A+POWER.
Ø38  16            ASL A
Ø39  ØD EE         BMI EXIT
Ø4Ø  FØ            ADD POWER
Ø41  ØD EB         BMI EXIT
Ø42  83            STA DENOM
;THEN TERM←1/DENOM WITH POINT
;ASSUMED TO BE LEFT OF TERM.
Ø43  12            ZER A
Ø44  82            STA TERM
Ø45  14            INC A
Ø46  Ø1 23         DIV TERM,
                       DENOM
;IF SIGN=Ø[255] THEN SUM←
;SUM+[-]TERM.  SIGN FLIPS.
Ø47  92            LDA TERM
Ø48  61            LDZ SIGN
Ø49  Ø9 Ø3         ZEQ DOWN1
Ø5Ø  15            NEG A
Ø51  1C            INC Z
Ø52  1C            INC Z
Ø53  1B      DOWN1 DEC Z
Ø54  51            STZ SIGN
Ø55  F1            ADD SUM
Ø56  81            STA SUM
;THEN COUNT←COUNT+1.
```

(Continued On Next Page)

```
Ø57  63              LDZ LCOUNT
Ø58  1C              INC Z
Ø59  53              STZ LCOUNT
Ø6Ø  Ø8 Ø3           ZNE DOWN2
Ø61  62              LDZ HCOUNT
Ø62  1C              INC Z
Ø63  52              STZ HCOUNT
;POWER IS UPDATED.
Ø64  32        DOWN2 JSR MULPWR
Ø65  ØD CF           BMI EXIT
Ø66  32        ENTER JSR MULPWR
Ø67  ØD CC           BMI EXIT
Ø68  62              LDZ HCOUNT
Ø69  54              STZ HIGH
Ø7Ø  12              ZER A
Ø71  63              LDZ LCOUNT
Ø72  Ø7 DØ           BRN BACK
;MAIN ARCTAN LOOP ENDS HERE.
;EXITS OCCUR WHEN IT'S CLEAR
;DENOM WOULD BE TOO LARGE
;FOR DIVISION IN LINE Ø46.
;
;IN MULPWR SUBROUTINE, POWER←
;POWER*5[239] IF FLAG=Ø[255].
;ADDING AND SHIFTING HERE IS
;FASTER THAN MULTIPLICATION.
Ø73  9Ø       MULPWR LDA POWER
Ø74  6Ø              LDZ FLAG
Ø75  Ø9 ØD           ZEQ MUL5
Ø76  Ø7 Ø5           BRN MUL239
Ø77  1C          TOP INC Z
Ø78  AØ              EXA POWER
Ø79  15              NEG A
Ø8Ø  EØ              SUB POWER
Ø81  AØ              EXA POWER
Ø82  16       MUL239 ASL A
Ø83  ØD ØD           BMI OUT
Ø84  16              ASL A
Ø85  ØD ØA           BMI OUT
Ø86  16         MUL5 ASL A
Ø87  ØD Ø7           BMI OUT
Ø88  16              ASL A
Ø89  ØD Ø4           BMI OUT
Ø9Ø  Ø8 ED           ZNE TOP
Ø91  FØ              ADD POWER
Ø92  8Ø              STA POWER
Ø93  11          OUT RTS
;NEXT SUBROUTINE CONVERTS
;BINARY FRACTION INTO DECIMAL
;FRACTION.  ACCUMULATOR, FOR
;INPUT, HOLDS STOP BIT AND ANY
;DESIRED LEADING DIGITS.
;MULTIPLICATIONS BY TEN (BY
;ADDING AND SHIFTING) YIELD
;DECIMAL DIGITS.  SECOND CALL
;TO SUBROUTINE WILL EXTRACT
;REMAINING SIGNIFICANT DIGITS,
;BUT TRAILING INSIGNIFICANT
;DIGITS WILL BE INCLUDED.
Ø94  82       CONVRT STA DECFRC
Ø95  93              LDA BINFRC
Ø96  16              ASL A
Ø97  Ø6 ØØ           LDZ #Ø
Ø98  ØA Ø2           BCC A1
Ø99  Ø6 Ø5           LDZ #5
1ØØ  17           A1 LSR A
1Ø1  83              STA BINFRC
1Ø2  16              ASL A
1Ø3  16              ASL A
1Ø4  ØA Ø2           BCC A2
1Ø5  1C              INC Z
1Ø6  1C              INC Z
1Ø7  16           A2 ASL A
1Ø8  ØA Ø1           BCC A3
1Ø9  1C              INC Z
11Ø  17           A3 LSR A
111  F3              ADD BINFRC
112  16              ASL A
113  83              STA BINFRC
114  ØA Ø1           BCC A4
115  1C              INC Z
;Z NOW HOLDS NEXT DIGIT.
116  92           A4 LDA DECFRC
117  16              ASL A
118  16              ASL A
119  16              ASL A
12Ø  16              ASL A
121  Ø9 Ø4           ZEQ A5
122  14         LOOP INC A
123  1B              DEC Z
124  Ø8 FC           ZNE LOOP
125  ØA D7        A5 BCC CONVRT
126  11              RTS
;
;END OF PROGRAM---162 BYTES
;WITH CHECKSUM=1ØØ97.
```

C

Defining the RS-232C

by Linda Lee

For those who make programming their business, the RS-232C is all in a day's work. But for others who use this kind of information every Friday during a full moon followed by a blizzard, the RS-232C can be synonymous with rising blood pressure and tension headaches.

To begin with, what is an RS-232C anyway? RS stands for Recommended Standard. This standard defines a set of connectors that are used to join terminals to modems or other external devices. These connectors contain 25 wires with 25 corresponding connector pins. Each pin and wire is given a specific function (see Figure 1). These permissible functions make up the Recommended Standard. The equivalent international standard is CITT V24.

All 25 pins are never used at one time. The needed functions are determined by the network designer. All other pins are ignored. Usually about six pins are used at any one time.

To understand how the hardware works let's look at a sequence of events that may take place when using a modem. The pins within the RS-232C require a signal, either "on" or "off", to talk to another device. When the pin is turned on by the computer the defined message is sent. When it is turned off no message is sent.

For example, pin 7 is the signal ground, which is the common pin for all signals. This will be turned on first. Next, Data Terminal Ready (DTR) at pin 20 must be on to signal that the terminal is powered up and ready. Then the Data Set Ready (DSR) at pin 6 must be on to signal that the modem is powered up and ready. Now everything is set, but no information is flowing in either direction.

At this point pin 4, which is Request To Send (RTS), is turned on. This signals the modem that the terminal wants to send data as soon as possible. When everything is set, pin 5, which is Clear To Send (CTS), is turned on. This tells the terminal that it can begin transmitting to the modem. Pin 2, which is Transmission Data, and pin 3, which is Data Received, will now be used to send and receive data to and from the terminal.

You can now see that by using these hardware standards, a universal "language" has developed.

The user port on the VIC 20 and the Commodore 64 does not have the RS-232C pin configuration. It can be easily converted to one, though, with a Commodore RS-232C interface, Part #1011A, or with a modem.

Now that you understand more about the hardware you must learn how to open an RS-232C channel to enable the hardware to do its job. This is done with interface software, which is accessed from BASIC or machine language. Within the RS-232C there are two software registers called the Control Register and the Command Register. Each register has a specific function for accessing the channels. Before we proceed any further let's clear up some definitions.

Control Register.

A single-byte character that specifies the baud rate. This register is required to open an RS-232C channel. Within this register are three parameters which must be set. They are:

Stop Bits: Certain computers need blank bits sent after each character to signal the end of a character. Normally, a 300 baud computer gets 8 bits of data for each character. This makes a total of ten bits, including the start and stop bit. Therefore, 300 baud becomes 30 characters per second.

Word Length: This controls how many bits are in each character. This can range from five to seven bits. Most computers use seven or eight.

Baud Rate: This is the speed of communication in bits per second. The VIC 20 and the C64 range from 0 to 1200 baud.

Command Register.

A single-byte character that specifies other terminal protocol. It is not required to open an RS-232C channel. There are three parameters within this register also. They are:

Parity: Some computers check for transmission errors by setting the highest bit in each character in a certain way.

Even Parity: The total number of "on" bits in each character should always be an even number.

Odd Parity: The total number of "on" bits in each

character should always be odd.
Mark Parity: The highest bit is always turned "on".
Space Parity: The highest bit is always turned "off".
No Parity: The parity is disabled. There is no parity generated or received.

Duplex:

Two lines that operate in the same way.
Full Duplex: This is the movement of data in two directions simultaneously when communicating with another device or computer. This can be compared to the way an echo bounces back. Full means that when the message is sent out on one line it gets to the receiving end and bounces back to the originator.
Half Duplex: This duplex does not echo. Data moves in a single direction at any time.

Handshaking:

An exchange of signals between computers or peripherals, such as modems, to determine whether a machine is ready to send or receive.
Three-Line Handshaking: This involves three lines—ground, transmission (for sending data) and receiving (for taking in data). It is generally not used with modems.
X- or Full-Line Handshaking: This involves pins 4,5,6,8 and 20 for clear to send and data terminal. It uses multi-lines for receiving and transmitting data. When the RTS (request to send), CTS (x-line) and DCD (receive line signal) are implemented, the VIC 20 and Commodore 64 become Data Terminal Devices. This is the common handshaking for modems and telecommunications.

Now that you have a better understanding of some things that take place within the RS-232C let's take a look at the normal register settings on the VIC 20 and Commodore 64.

The baud rate is the only thing that doesn't have a preset condition. So whenever you access the RS-232C this must be set. Therefore you will always have a value in the Control Register (Figure 2). The other preset Control Register settings are one stop bit and eight bits data word length.

Within the Command Register (Figure 3), the preset conditions are: no parity, full duplex, and a three-line handshake. Obviously this protocol won't always work for your particular needs. Changing the settings are simple once you understand some binary functions.

In Figures 2 and 3 you will notice boxes containing numbers from seven to zero with lines connecting them to options within the registers. These boxes represent the powers of two. For example, the box with a four inside would be 2↑4 = 16 and the box with a zero inside would be 2↑0 = 1. The bits from seven through four are the high bits, and from three through zero are the low bits. The values of these bits in descending order are 128, 64, 32, 16, 8, 4, 2 and 1. It is a good idea to write these values above the boxes.

Since binary numbers work on base two, there are only two digits involved (all bases range from zero to a number equal to the base less one) namely zero and one. Zero will turn off a bit and one will turn it on. Within the Command Register, you can see on the map that no parity, full duplex, and a three-line handshake all have bits set at zero. Since there are no bits turned on there is a zero value for this register at the normal setting of the computers.

In order to change any register value you must first determine what setting you need to change it to. This can be done by asking the telecommunications service you wish to contact or by checking your printer manual for the baud rate, stop bits, word length, parity, duplex and handshake. Then you check the maps to determine which bits will be turned on with those settings. If you need odd parity, half duplex and an x-line handshake, you will be turning on bit five, bit four and bit zero. Here is where the powers of two are needed. Follow the lines from your selections to the bit values and make your calculations. The values are added to determine the register value.

2↑5 + 2↑4 + 2↑0 = 32 + 16 + 1 = 49

The Command Register value would be 49.

Let's try another one for the Control Register. If you need one stop bit, seven bits data word length and a 300 baud rate, you will be turning on bit five, bit two and bit one. In this case one stop bit has a value of zero, which doesn't turn on the bit. Therefore, you don't need to calculate for it.

2↑5 + 2↑2 + 2↑1 = 32 + 4 + 2 = 40

The Control Register value would be 40.

Here's one more for good measure: two stop bits, five bits data word length and 1200 baud for the Control Register. Bit seven, bit six, bit five and bit three are turned on.

2↑7 + 2↑6 + 2↑5 + 2↑3 = 128 + 64 + 32 + 8 = 232

The Control Register value would be 232.

In the Command Register you need mark parity, full duplex and an x-line handshake. Bit seven, bit five and bit zero are turned on.

2↑7 + 2↑5 + 2↑0 = 128 + 32 + 1 = 161

The Command Register value would be 161.

Great! Now that you have that straight, what do you do with it? This depends on what you are doing with the RS-232C. If you are using an RS-232C third party printer with software, you would have to follow the directions within the software package for making changes. But if you are printing in a direct mode, you need the following BASIC syntax to open the channel:

```
open lfn,2,2,chr$(Control Register Value),
        chr$(Command Register Value)
```

Lfn is the logical file number, which can be any

number from one to 255. But if you choose a number greater than 127, an automatic line feed will follow all carriage returns. Use this carefully!

The first two is the device number. When addressing an RS-232C device, it is always known as device number two.

The second two is the secondary address or command channel.

The Control Register value comes first. Since the baud must always be set there will always be a value in this register.

The Command Register value comes last. Since there can be a value of zero in this register, it is not always needed.

When you are using modems, the changes are usually done within a menu of the terminal software. Usually the changes can be made without making any calculations. This is by far the easiest way to access a channel.

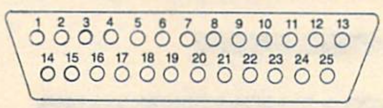
The RS-232C can be a useful tool for exploring the vast possibilities of telecommunications as well as communicating with certain external devices. Everyone can't be a programmer, but everyone can learn to use RS-232C to fit their needs. **C**

Figure 1.

Mode:
1....3 line (S in, S out, GND).

2....X line
(Note) *These lines will be kept HIGH during 3-line mode.
**Jumper is not connected.
***For only VIC 1011B CURRENT LOOP TYPE.

RS-232C CARTRIDGE CONNECTOR



PIN NO.	SIGNAL	EXPLANATION	ABV.	EIA*	SIGNAL DIRECTION	MODE
1	GND	Ground				1 2
2	SD	Transmitted Data	RTS	BA	Output	1 2
3	RD	Received Data		BA	Input	1 2
4	RS	Request To Send	CTS	CA	Output	1* 2
5	CS	Transmission is Possible	DSR	CB	Input	2
6	DR	Data Set Ready		CC	Output	
7	GND	Signal Ground	DCD			
8	CD	Carrier Detect		CF	Output	2
9	CL+	Current Loop+				**
10	CL−	Current Loop−				**
11-19		No Connection	DTR			
20	ER	Data Terminal Ready		CD	Input	1*
21-25		No Connection				

*EIA: Electronic Industry Association

Bit Value 128 64 32 16 8 4 2 1
7 6 5 4 3 2 1 0

Stop Bits

0-1 Stop Bit
1-2 Stop Bits
1 Stop Bit If Word Length = 8 Bits & Parity
1½ Stop Bits If Word Length = 5 Bits & No Parity

Word Length

Bit 6	Bit 5	Data Word Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

Unused

Baud Rate

3	2	1	0	Baud Rate	
0	0	0	0	User Rate	
0	0	0	1	50 Baud	
0	0	1	0	75	
0	0	1	1	110	
0	1	0	0	134.5	
0	1	0	1	150	
0	1	1	0	300	
0	1	1	1	600	
1	0	0	0	1200	
1	0	0	1	1800	[NI]*
1	0	1	0	2400	[NI]
1	0	1	1	3600	[NI]
1	1	0	0	4800	[NI]
1	1	0	1	7200	[NI]
1	1	1	0	9600	[NI]
1	1	1	1	19200	[NI]

*NI—Not Implemented

Figure 2. Control Register Map

Bit Values 128 64 32 16 8 4 2 1
7 6 5 4 3 2 1 0

Parity Options

Bit 7	Bit 6	Bit 5	Operations
-	-	0	Parity Disabled, None Generated/Received
0	0	1	Odd Parity Receiver/Transmitter
0	1	1	Even Parity Receiver/Transmitter
1	0	1	Mark Transmitted Parity Check Disabled
1	1	1	Space Transmitted Parity Check Disabled

Handshake

0-3 Line
1-X Line

Duplex

0-Full Duplex
1-Half Duplex

Unused

Unused

Unused

Figure 3. Command Register Map

Creating Reports with the Commodore 64 *Manager*

by Thomas Ziegler
Commodore Software

Many first-time users of the Commodore 64* Manager *data base package are somewhat bewildered by the myriad of options and features that this program offers. One area in particular, Report Generate, can be confusing, particularly to a novice. To help you allay the confusion, Tom explains how to use the* Manager's *powerful report generator to produce sorted reports.

To keep the explanations simple, I will describe a simple mailing list application and show you how to create two reports: a report sorted by last name and a report sorted by last name that also uses the *Manager's* search criteria. These two reports should provide you with a good introduction to the *Manager's* report generator. Figure 1 shows the record layout used for the mailing list file, with each field numbered from one to 13.

Before you even turn on your computer, you should first lay out the report on paper, indicating exactly where each piece of information will go. Using grid paper makes designing a report easier because you can indicate the spacing more precisely. When you lay out a report, you must define every single piece of information that is going to appear on the report.

Figure 2 depicts the first report: an alphabetical list by last name. If you look at this report layout, you can see that at the top of each page of the report, as a heading, we want the title of the report centered. Centered below the title is the current date. We then want to skip a space and begin printing the actual data for the report. For each person on the mailing list, we want to show last name, first name, address, city, state, zip code, telephone number and membership status. We also want to skip a line between each person. Finally, at the bottom of each page, we want to put a page number. For this particular report there will be no column headings because all the information on the report is self explanatory (except possibly membership status). Eliminating column headings allows us to get more records per page, saving paper. (On the second report, however, column headings will be used.)

From this description, you can see that there are basically three parts to the report: the heading, the data from the records and the footer (i.e., the page number). The *Manager's* report generator calls each one of these parts of a report a print zone. In the *Manager,* there is a header zone, a list zone (for the data) and a footer zone. For each of these zones you have to define what data will be printed for the zone. Defining the data to be printed is done by setting up print areas in the *Manager's* re-

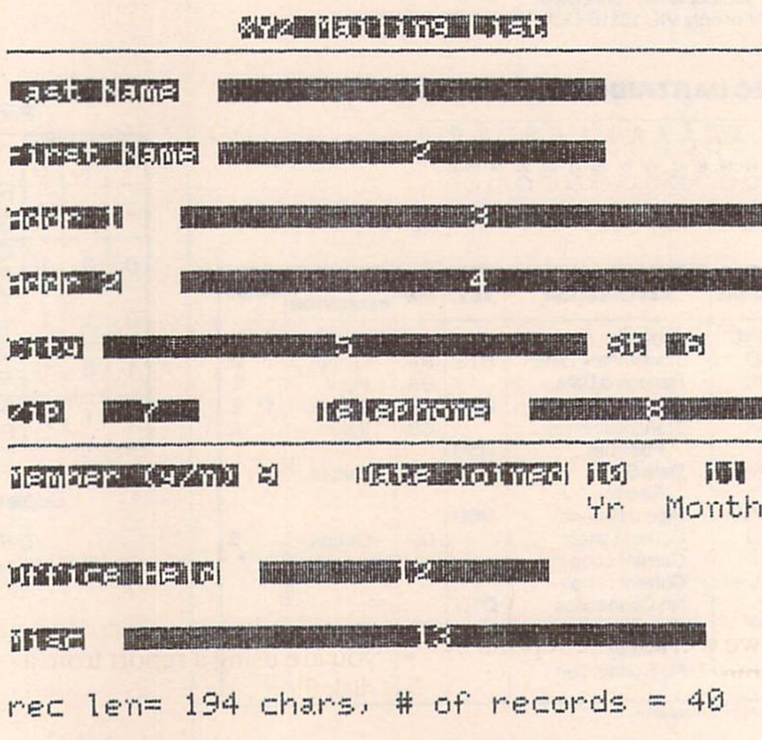


Figure 1. Layout for Each Record

XYZ MAILING LIST
AS OF: CURRENT DATE

LAST NAME	FIRST NAME		MEMBER STATUS	
ADDRESS 1	ADDRESS 2			
CITY		STATE	ZIP	TELEPHONE
LAST NAME	FIRST NAME		MEMBER STATUS	
ADDRESS 1	ADDRESS 2			
CITY		STATE	ZIP	TELEPHONE

Figure 2. Setting Up Your Report

port generator. In addition, you also have to consider the following items:

1. Search characteristics. The *Manager's* report generator has the ability to be selective in its reports. You can print a report based on exact matches, hunts or complex combination searches. The alphabetical list of our mailing list does not require any searching. We are simply listing everyone in the file. However, the second report, which lists only members, will take advantage of the *Manager's* search abilities.

2. The order of the report. The *Manager* provides three ways to sort or order a report: entry (file) order, index field order, or single or multiple combination sorts based on any field. Entry-order reports list the data records in the order that they were entered. This type of report is the fastest to produce. Index-field-order reports use the index created in the Enter/Edit option as the basis for the report. Reports sorted by index are very fast. Reports sorted by fields are the most powerful and flexible. However they are also the slowest, since the file has to be sorted before the report can be printed. In our example reports, we will sort the reports by last name.

3. Will the report be printed, displayed or stored on disk? The *Manager* allows you to produce reports for the screen or for the printer. It also allows you to direct a report to the disk drive. Reports to disk can be used by *Easy Script* as fill files.

Once you have entered all the requirements for your report, the *Manager* permits you to save your report specifications on disk for later use. The next time you want to produce this report, all you do is tell the *Manager* which report you want and it will automatically start printing.

Let's look more closely at the steps you need to follow to produce the two reports we defined above.

Alphabetical Report By Last Name

The format for this report, as shown in Figure 2, assumes that you have a printer. Here is how to produce it.

From the main menu of the *Manager,* select R for the Report Generate option. The first prompt is REPORT FROM KEYBOARD OR DISKFILE? The first time you define a report, you are defining it from the keyboard by typing in the report specifications. After the report has been defined and the specifications are saved on a disk, you are using a report from a disk file.

You next have to provide the name of the file you are using for your report.

The next prompt, ENTER SEARCH CRITERIA, is used to select specific records. In this example there are no search criteria required, so you would press RETURN.

Next you have to specify the order of the report: index, sort or file. In this case we want a sorted report. After you decide you want a sorted report, you have to define the sort. To help define the sort, the *Manager* prompts: ENTER NUMBER OF SORT KEYS. A key is simply the field or fields we are using in the sort. The *Manager* allows a maximum of 16 keys. In this case, we only need one key, last name. When the *Manager* sorts the data for this report, the data will be sorted by last name.

After all your typing, the *Manager* screen should look like Figure 3 on page 100. At this point, you have to specify which fields you want to use in the sort. In this case, last name is field one. The prompts shown in Figure 3 are used to specify whether the sort is alphanumeric or numeric and whether the data should be sorted in ascending or descending order. At these prompts, you enter your information and press the arrow-left key to continue.

Next, you have to specify where you want your output to go. If you recall, output can go to the screen, the printer or to a disk. Figure 4 on the next page shows the prompts that must be answered. In our example, we want an output to the printer.

want an output to the printer.

The next step is defining the characteristics of the three print zones: header, list and footer. For the sample report, you must first define the header zone. Checking our report layout in Figure 1, you can see that you need to allocate three lines for the header: one line for the title, one line for the date and one blank line to separate the heading from the rest of the report. Consequently, at the next prompt, NUMBER OF LINES IN ZONE? you would type "3". The next prompt is NUMBER OF LINES FROM THE TOP? By changing this number, you can vary the top margin of your report.

Next, the print-position editor is used to lay out your report. The print-position editor is used in all three zones. Figure 5 shows a blank print-position editor screen and provides explanations of each entry on the screen.

It is important to remember that every item (even spaces or blank lines) that you want to appear on a report must be specified in either the header, list or footer zone. Data specified in the header or footer zone will be printed once per page. Data specified in the list zone will print once per record. All the entries in the print position editor are not used for every print area that you specify. For example, if you enter a text data type, no subscript entry is required.

To enter the print areas for each zone, you simply type in the required data and press <RETURN>. The data for our example report is shown in Table 1 on page 102. Listed below are keys that perform special functions in the *Manager's* print-position editor.

f7: Move to the next print area in the zone.
f8: Go back to the previous print area in the zone.
CRSR UP: Move to the previous item on the screen.
CRSR DOWN or RETURN: Move to the next item on the screen.
SHIFT CLR/HOME: Delete the print area displayed on the screen.

```
                 report generate
                 sort conditions

enter number of sort keys? 1

field    len   alpha/num          ascd/desc
 1       20        a                  a
```

Figure 3. Prompts Used to Specify Report Fields

```
                 report generate
               output conditions

output to screen, printer, disk? p

enter device number? 4

enter printer control character? 0

enter line length? 80

enter number of lines per page? 66
```

Figure 4. Prompts for Report Output

```
                report generate
                  header zone
print area #  1              10 areas open

data type (t-n-f-d): f    subscript: 1
text/title

.........1.........2.........3.........4
length of area: 20  line number     : 1

column number : 1   centering (y/n)?  n

# of decimals  :    accumulate (y/n)? n

break type (none, infds, page)? n
```

Figure 5. Print-position Editor

Explanation of Prompts

Print Area. Each data item displayed in a report will automatically be assigned a print-area number. You press f7 to advance to the next print area.

Data Type — F R D T. There are four data types available for reports: fields, registers, display areas and text.

Subscript. When used with the field data type, this entry designates the field number. When used with the register data type, this entry designates the register number.

Text/Title. Used to enter the text for a text data type. Whatever is entered here will be displayed as entered in the report. When used with a field or register data type, data entered here will be a column heading.

Length of Area. Defaults to the field size if the field data type is used. If the register data type is used, it defaults to 12. The number entered here may be changed to be less than the default.

Line Number. Designates which line in the particular print zone this data item will be placed in.

Column Number. Defaults to the next available horizontal position based on the previous entry. It may be modified.

Centering Y/N. When a Y is entered here, the data will be centered in the area specified.

Number of Decimals. Used for numeric fields or registers to specify the number of decimal places to be displayed.

Accumulate Y/N. If Y is entered here, totals for this column will be displayed at the end of the report or at break points.

Break Type. If a break point is specified, the breaks can be linefeeds or a new page.

XYZ MAILING LIST
AS OF: CURRENT DATE

Clark	Gail	Y
611 Windsor Place		Apt. 312
Anytown	PA 19000	215-989-4321
Fretz	Dr. Walter	N
1101 Walnut Street		
Anytown	PA 19000	215-989-6789
Jones	Susan	N
1525 Ridge Road		
Pinebrook	NJ 07005	609-123-4567
Smith	Jim	Y
421 Broad Street		
Anytown	PA 19000	215-989-4653
Smith	Joe	Y
11 Winding Road		
Anytown	PA 19000	215-987-1212
Smith	John	Y
123 Main Street		
Anytown	PA 19000	215-989-4567

Figure 6. Sample Report of Mailing List, Alphabetized.

f1: Return to the main menu.
f2: Return to the beginning of the Report Generate option.

After you enter data for your first print area, you have to press f7 to enter the information for the next print area. Each time you finish with one print area, you have to press f7 for the next print area. Pressing f8 allows you to go back and edit a previous print area. When all the print areas for a zone are entered, you have to press the arrow-left key.

When you finish entering the print areas for one print zone, you then follow the instructions on the screen and continue entering data for all the remaining print areas. Finally, you exit the print-position editor.

If you look at Table 1 for the footer zone, you will notice that the register data type is used with subscript 104. Register 104 is a special register that is used to keep track of the page number of your report.

Next, you are asked if you want to save the report conditions. For a sorted report, you must save the report conditions. Otherwise, the *Manager* will prompt you to insert your data disk over and over again. After the *Manager* saves your report specifications on disk, you can use the report specifications to print additional reports.

When the *Manager* prompts: SORT THE FILE? if you enter Y, the *Manager* will sort your file and prompt you to alternately insert the program and data disks. After the file is sorted, the report will be printed.

If you enter N at SORT THE FILE, the *Manager* will check your data disk to see if you have already

produced a sorted report with your file. If you have, the *Manager* will use the sort file from this previous report rather than creating a new sort file.

Once the report is printed, you have the option to print it again or return to the main menu. Figure 6 on page 101 shows a sample report produced from the specifications just described.

Members-Only Report

This report is similar to the alphabetical list report except that we are going to add column headings and use search criteria to limit the number of records printed in the report. Figure 7 shows the report layout. As you can see, this report is simpler than the alphabetical list report.

The search criteria are fairly simple: we want to list only members. In terms of our file, this means we want to list only people who have a "Y" entered for field nine. When the *Manager* prompts SEARCH CRITERIA, we will enter: F9="Y". This means include only records with field nine equal to "Y" in this report. Any record without a "Y" in field nine will not be included in the report.

The sort specifications for this report are identical to the sort specifications used in the previous report. Table 2 shows the print areas used in this report. Notice that the column headings are part of the header zone.

Remember that you have to press the f7 key to advance to the next print area you wish to enter. The arrow-left key is used to proceed to the next print zone after all the print areas for a particular zone are entered. Also, remember that sorted reports must be saved before they can be printed. Figure 8 shows a sample report.

Once you master creating simple reports like the two described in this article, you can go on to create more advanced reports that use arithmetic and registers. The possibilities are virtually limitless.

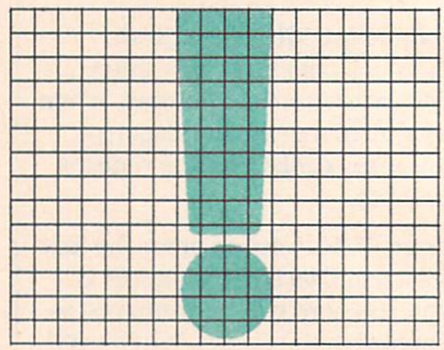


Table 1. Data for Alphabetical Report

The data in the name column is used for the TEXT/TITLE for text data types.

PRINT AREA	NAME/TEXT	DATA TYPE	SUBS.	LENGTH	CENTER	LINE	COL.
Header Zone: 3 lines in the zone							
1	XYZ MAILING LIST	T	—	80	Y	1	1
2	As of: Current Date	T	—	80	Y	2	1
3	Space	T	—	1	—	3	1
List Zone: 4 lines in zone							
1	Last Name	F	1	20	—	1	1
2	First Name	F	2	20	—	1	21
3	Member?	F	9	1	—	1	41
4	Address 1	F	3	30	—	2	1
5	Address 2	F	4	30	—	2	31
6	City	F	5	25	—	3	1
7	State	F	6	2	—	3	27
8	Zip Code	F	7	5	—	3	30
9	Telephone	F	8	12	—	3	40
10	Space	T	—	1	—	4	1
Footer Zone: 1 line in zone							
1	Page #	R	104	4	—	1	40

	XYZ PHONE LIST MEMBERS		
NAME		DATE JOINED	PHONE #
FIRST NAME	LAST NAME	MONTH/YEAR	XXX-XXXX

Figure 7. Setting Up Member Phone List

XYZ PHONE LIST MEMBERS			
	NAME	DATE JOINED	PHONE #
Adams	John	10/81	215-123-4567
Brown	William	11/82	215-123-5432
Caldwell	David	05/83	215-432-9876
Dawkins	Joseph	04/83	215-123-1234

Figure 8. Sample Report of Member Phone List

Table 2. Data for Members-Only Report

The data in the name column is used for the TEXT/TITLE for text data types.

PRINT AREA	NAME/TEXT	DATA TYPE	SUBS.	LENGTH	CENTER	LINE	COL.
Header Zone: 6 lines in the zone							
1	XYZ MAILING LIST	T	—	80	Y	1	1
2	MEMBERS	T	—	80	Y	2	1
3	Space	T	—	1	—	3	1
4	Name	T	—	10	—	4	8
5	Date Joined	T	—	11	—	4	42
6	bbPhone #	T	—	9	—	4	64
7	-------... (40 –)	T	—	40	—	5	1
8	-----------	T	—	11	—	5	42
9	-----------	T	—	12	—	5	64
10	space	T	—	1	—	6	1
List Zone: 2 lines in zone							
1	First Name	F	2	20	—	1	1
2	Last Name	F	1	20	—	1	20
3	Month Joined	F	11	2	—	1	44
4	/	T	—	1	—	1	45
5	Year Joined	F	10	2	—	1	46
6	Telephone	F	8	12	—	1	64
7	Space	T	—	1	—	2	1
Footer Zone: 1 line in zone							
1	Page #	R	104	4	—	1	40

C

Adding Power to Your *Word Machine*

by Philip S. Dale

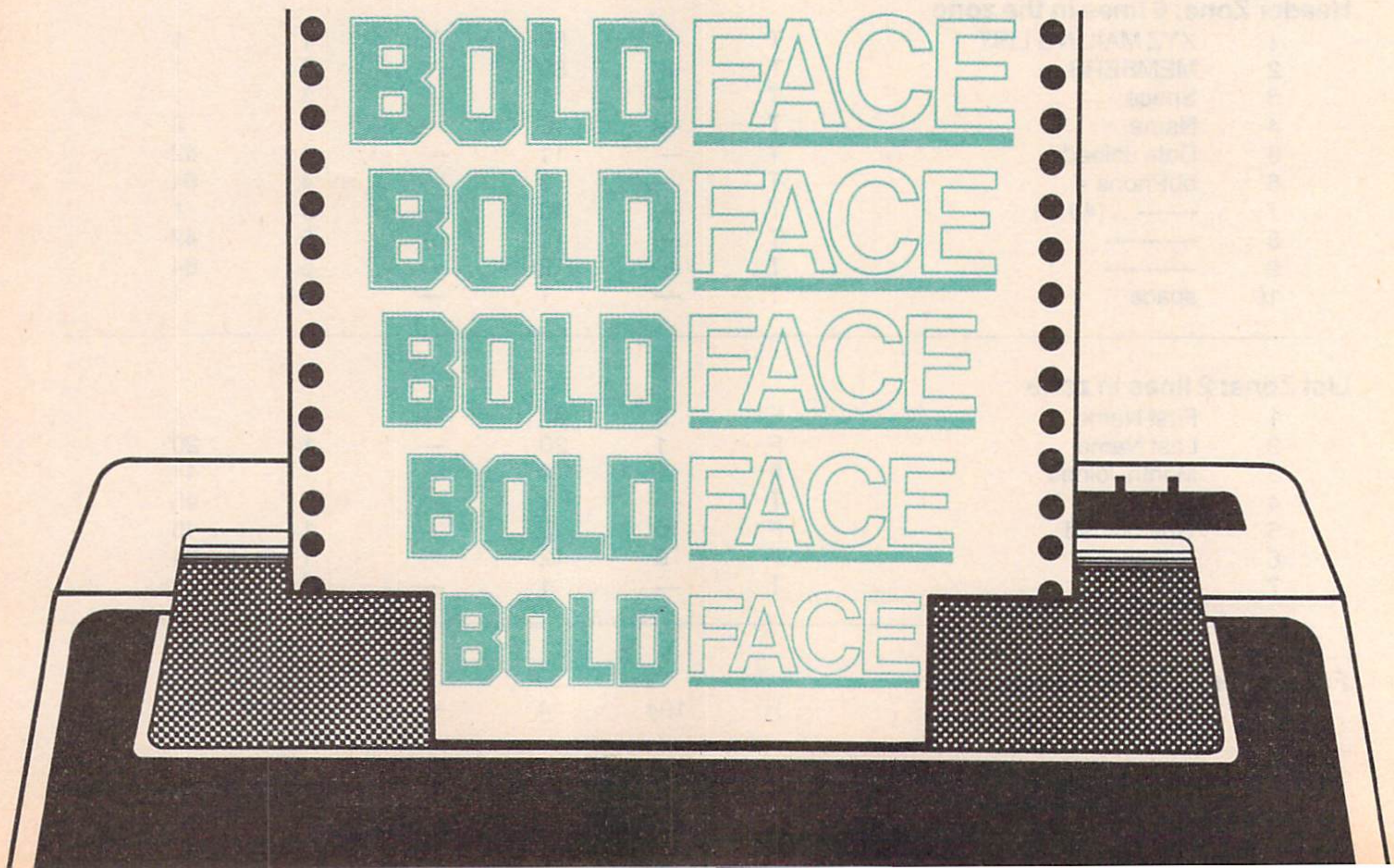
The *Word Machine* is an inexpensive and easy way to begin word processing. It is easy enough for children to use, and has enough features—insertion and deletion, search-and-replace, tape and disk storage, easy interfacing with the *Name Machine* mailing list program—to make it satisfactory for a wide range of correspondence and other home uses. It is unusual among word processors in that it is written in BASIC. This makes it somewhat slower in operation than machine language programs, but it has the considerable advantage of making the program easy to modify and "custom tailor."

There is now a new generation of high speed, moderately priced dot-matrix printers with great flexibility in printing style. My ProWriter has, among other options, the ability to print in either single-strike or "boldface" (double-strike), and I generally prefer the latter because it is nearly letter quality in appearance. It also can print underlined characters as a single unit.

Although the *Word Machine* does not have the built-in capability to use the double-strike and underlining capacities of my printer (there is an underlining option, which operates by the awkward device of backing up and printing the underline in a second pass, but it does not actually print the line *under* the character), I discovered how you can add these features yourself by making several simple changes in the *Word Machine* listing.

First, check your printer manual for the characters that need to be sent to the printer to select boldface or standard print. For the ProWriter, they are CHR$(27);CHR$(33) and CHR$(27);CHR$(34), respectively. Then add the following lines to the *Word Machine* program, using your printer's character codes in place of the ProWriter codes (you will need to be in upper/lower case mode for this):

```
7032 PRINT:PRINT"IS DOCUMENT TO
     BE PRINTED IN"
```



```
7033 INPUT"[RVS]B[RVOFF]
     OLDFACE OR [RVS]S[RVOFF]
     TANDARD TYPE";RR$
7035 IF RR$<>"B"AND RR$<>"S"THEN
     GOTO 7032
7037 IF RR$="B"THEN PRINT#1,
     CHR$(27);CHR$(33):GOTO 7040
7038 IF RR$="S"THEN PRNT#1,
     CHR$(27);CHR$(34)
```

Note that lines 7037 and 7038 will be different if your printer uses different control codes.

Next, check the manual for the characters that turn the underlining function on and off. For the ProWriter, they are CHR$(27);CHR$(88) and CHR$(27);CHR$(89), respectively. These codes must be inserted in the string that is sent to the printer (P$). However, there is an additional problem, since the CHR$ values are between 65 and 90. My printer interface from Cardco, as part of the Commodore-to-standard ASCII translation, subtracts 32 from all codes in this range. For a Cardco interface, the solution is to use CHR$(216) and CHR$(217), since in this range, 128 is subtracted during the translation. If you are using a different interface or a software driver, check the table of translations to see how best to get the values 88 and 89 delivered. Then add the following lines:

```
7735 CC$=""
7740 IF C=91 THEN CC$=CHR$(27)
     +CHR$(216)
7745 IF C=93 THEN CC$=CHR$(27)
     +CHR$(217)
7750 P$=LEFT$(P$,I-K)+CC$+MID$
     (L$,I+1)
7760 IF C=91 OR C=93 THEN K=K-2
7770 RETURN
7930 IF F THEN PRINT:PRINT#1,P$;
```

In this case, lines 7740 and 7745 may be different, depending on your printer and interface.

Now try the program, using the various options. If you are having problems with spacing (getting double spacing instead of single, and triple spacing instead of double), add the following line:

```
7020 OPEN 1,4,1:F=1
```

Be sure to save this modified program under a new name (perhaps "WORDMACHINE PLUS"), so that you will have it available, as well as the original *Word Machine* program. C

Creating Control Keys on the 8032

by Joe Rotello

Somewhat similar in function to the separate control keys on computers that use CP/M™ as an operating system, a CBM control key is a key (it can be any convenient key) that is set aside to activate specific functions or "control applications". For example, a program might be exited or a delete executed when you press the control key and a second key. The second key—we term it the "action" key—would be among a list of one or more keys in a program that activates a specific action (like "delete", or "show next record") when it is pressed immediately after the control key is pressed. Pressing the control key merely serves to alert the program that the next key pressed will be an action key.

On the CBM, we can define a short control key routine that can be made part of an overall larger program. It will achieve the same effect as having a predefined, separate "control key" present on the CBM keyboard.

Why Control Keys?

One of the main reasons behind the need for control keys lies in actual everyday programming hazards. Let's say you have created a really useful program. The main thrust of the program is that you want the video screen to be easy to read and understand, but at the same time you want to prevent the user from making mistakes as much as possible.

Now let's assume that the video screen in question displays information. At the bottom of your screen you have the simple instructions:

ENTER (I) To see Invoice (E) To Exit (D) To Delete Transaction

Ok, the user sees the simple instructions. But suppose the user selects "D", mistaking its location on the keyboard for the "E" key.

Oh my gawd! They really intended to press "E"!

Unfortunately, the program does what it is instructed to do and, assuming that the program does not happen to have any more failsafes, might proceed to erase or delete the transaction.

Oh, oh, problem city.

Now let's go back and see what might have happened if a control key system were in use. The user, in order to properly activate one of the instructions at the screen bottom, *first* presses the control key. No other key is pressed yet. The control key program proceeds to "light up" the control key indicator on the screen. If there is an error, "Hmm..." the user says, "I am in the control key mode and I don't want to erase that transaction record!" and they (a) hit the control key again, thus exiting control mode or (b) take the time to carefully select the proper action key.

So, the program has saved a mistake from happening. How?

1. The user has had to press *two* keys to execute a program instruction, first a "control" and then an "action" key.

2. The user *sees* the choice they selected (control mode) actually "light up" and has the opportunity to judge if the selection was proper.

3. If the selection was *not* proper, the user exits a potentially bad situation by merely pressing the control key again. The control key indicator is turned off, thus giving a positive indication to the user that they are out of control mode.

4. If the user *has* selected the proper function, they merely have to enter the appropriate action key to proceed.

The program included below is only one example of how you can implement a control key in your programs. Our particular program defines the ESC key as our control key, but you can define a different key, if you like. Our program also uses POKEs to define screen windows on the 8032. (For more information on windows, refer to my article on 8000 series windows in Issue 28.)

After running a little "do something" routine, the program GETs a keyboard entry and examines it to see if the key was the control key (ESC). If the key struck *was* the control key, the program jumps to a subroutine that lights up the control indicator and awaits the NEXT keyboard input.

If the next keyboard input is either another control key or *not* one of the action keys, the control action is terminated and the program returns to the main routine to begin looking at what may be input next.

If, however, the next key struck is an action key, then the program branches off to perform the function assigned to that action key. Frankly, our little example program does not produce a great deal of "action", but you get the picture.

This "on/off" control key action allows for "control key toggle". In this respect, a user may press the control key, perhaps decide that was *not* what they intended to do and, by pressing the control key again, abort the control subroutine and return program control to the main program logic. In addition, having the control indicator "light up" allows the program to verify to the user that the control key was indeed pressed and that the program is now waiting for further instructions, by way of more keyboard input from the user.

Control Key Uses

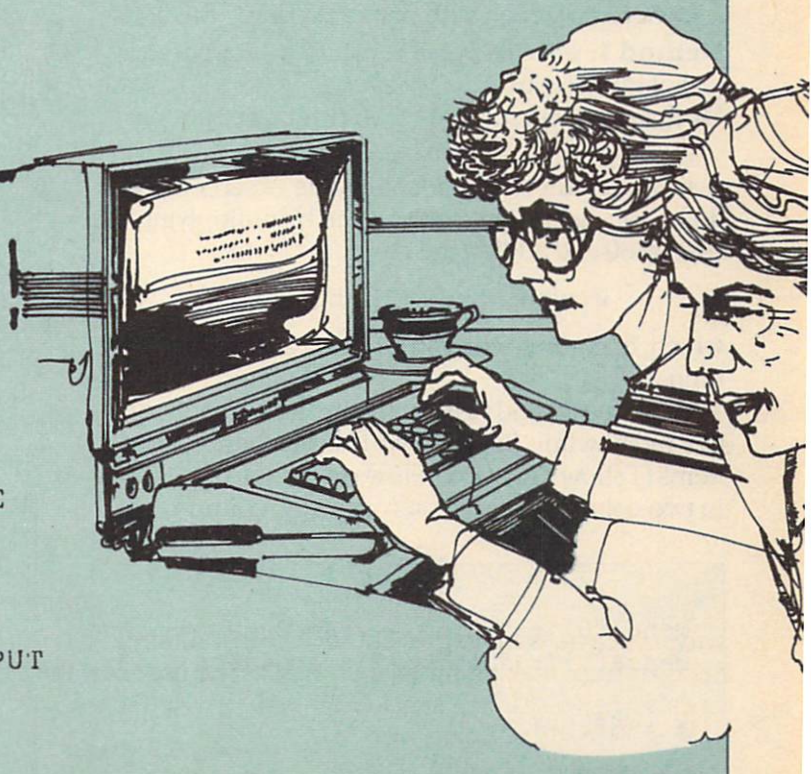
Control keys can be implemented in just about any type of computer program. You already see them in some of the professional CBM word processing software such as *Wordcraft*™ or *WordPro*™. Text, database or other programs where certain user errors or judgment mistakes really would be a disaster are prime candidates for control keys. So are programs where more than just a few keys on the 8000 keyboard would have to do double duty.

Programs using control keys can make use of the keyboard in other ways that many users might not think of. For example, terminal or other modem communications programs make very good use of control keys. The end product is a much more flexible computer program and system. **C**

Joe Rotello is a regular contributor to Commodore Microcomputers.

Control Key

```
10 REM    CONTROL KEY SIMULATOR
20 REM    FOR 8000/9000/SERIES
30 :
40 D$=" CONTROL[SPACE4]E=EXIT
   [SPACE5]Q=QUIT"
50 :
60 REM   PRINT D$ IN LOWER WINDOW
70 POKE 224,21:POKE 225,24
   :PRINT"[CLEAR]";D$
80 :
90 REM   REVERT TO UPPER WINDOW
100 REM  PRINT"[CLEAR]";
110 POKE 224,0:POKE 225,10
    :PRINT"[CLEAR]";
120 :
130 REM  PERFORM SOMETHING TO
    SIMULATE ACTION
140 FOR I=1 TO 50:PRINT I,
    INT(I^2),INT(I^3):NEXT
150 :
160 REM    GET USER INPUT
170 GET A$:IF A$=""THEN 170
180 :
190 REM    DEFINE <ESC> KEY AS
    CONTROL KEY
200 IF A$=CHR$(27)THEN 250
210 PRINT A$;:GOTO 170
220 :
230 REM   CONTROL KEY PRESSED,
     MOVE TO LOWER WINDOW
240 REM   LIGHT UP CONTROL LINE
250 POKE 224,21:POKE 225,24
    :PRINT"[CLEAR]";
260 PRINT"[RVS]"+D$+"[RVOFF]"
270 :
280 REM   GET NEXT KEYBOARD INPUT
290 GET A$:IF A$=""THEN 290
300 :
310 REM   ALLOW CONTROL TOGGLE
320 REM   IF <ESC> AGAIN OR NOT
    ACTION KEY(S) (Q OR E)
330 REM   TURN OFF CONTROL,
     LOOP BACK TO RUN AGAIN
340 IF A$=CHR$(27)OR A$<>"Q"AND
    A$<>"E"THEN PRINT"[CLEAR]";D$
    :GOTO 110
350 :
360 REM   IDENTIFY WHAT ACTION
    KEY WAS STRUCK
370 IF A$="Q"THEN PRINT"[CLEAR]
    YOU PRESSED CONTROL QUIT":END
380 IF A$="E"THEN PRINT"[CLEAR]
    YOU PRESSED CONTROL EXIT":END
```



SuperPET Potpourri

by Dick Barnes

In Issue 27 of this magazine, you may have read with interest David Bull's article on absolute cursor control in BASIC. (With absolute control you may put the cursor anywhere on the screen *without* reference to its previous location.) As I show below, absolute control is built into all languages in SuperPET. As a bonus, you'll find out how to use the assembler and linker in SuperPET to create a machine language program that will "retrieve" both a language and an otherwise lost program when you, by accident, forget to save the program to disk and leave the language. (It happens!)

Absolute Cursor Control

SuperPET tracks current cursor location at $122; you may sense that position at any time or command the cursor to appear at any spot on the screen. MicroBASIC uses one method to do this; the other languages employ a second. Let's look, first, at how it's done in mBASIC, where screen positions start at one, which is HOME, and end at location 2000, which is row 25, column 80. You send the cursor anywhere on screen with one of two basic methods:

Method 1: x=cursor(241) : print "We start on row 4 of the screen."

Method 2: if cursor(241) then print "We start on the same row."

If you'd rather not refer to a table of screen positions, you can assign cursor value by multiplying the row by 80 and adding the column:

if cursor (12*80+21) then print...

which, of course, will start printing at row 12, column 21.

You may tab and print with absolute cursor control most swiftly. Suppose you have a menu of ten items (I show only two below) to print to the screen in two columns, starting at row three, column one:

```
menu$(0) = "Enter New Accounts  = 0"
menu$(1) = "Update Old Accounts = 1"

a = 3*80+1
```

```
for i = 0 to 9
  if cursor(a) then print menu$(i)
  a = a + 40
next i
```

Suppose you want to print exactly 40 values of several hundred in an array (called "number") of numeric values of six digits or less, starting at a point in the array which you select:

```
input "Select starting value to
  print: ", start
a = 4*80+1
while cursor(a)
  print number(start) : a=a+10
  start=start+1 : count=count+1
until count = 40
```

Next, let's use absolute control to sense cursor position in a most useful way. We write a program that can be interrupted at any time with the OFF key, so the user can take a break or answer the phone. We must print a prompt at HOME to show the program has been interrupted, and when we restart it, we must resume printing on screen exactly where we stopped. There's a trick to it: x=cursor(0) assigns the current cursor location to variable "x". Pretty obviously, x=cursor(x) will put the cursor back where it was. In the example below, H$ homes the cursor, EL$ erases to end line, "!" marks a comment, and the code or ordinal (as Waterloo calls it) returned by the OFF key is 255.

```
... loop
...    ...            ! Merrily printing
       away to screen.
...    ...
100    get cease      ! No effect
       unless OFF has been pressed.
110    if cease=255
120      x=cursor(0) : print H$; EL$;
```

```
"Press OFF to resume: ";
130     get start : if start <>
        255 then 130
140     x=cursor(x)
150   endif
...   ...                ! Resume printout.
...   ...
   endloop
```

The virtues of structure show in this excerpt, which is from my library. I use the method in almost every program because I'm often interrupted by visitors and phone calls.

Inverting Control—Forbidden Zones

You may take the next technique as far as you like, even to the creation of zones which the cursor is forbidden to enter. We must use two simple algorithms, which convert cursor position to row and column. If "x" is cursor position in the 1-to-2000 set of values, then:

```
row = ip((x-1)/80)+1    and
    column = mod((x-1),80)+1
```

In microBASIC, the integer value of a division is "ip", or "integer part." The "mod" produces the *remainder* after x is divided by 80. Don't let "mod" throw you; 12 modulo 10 = 2, for 12 divided by ten leaves a *remainder*, after division, of two. We write the process as: "mod(12,10)". Having this in hand, let's irrevocably set the right margin on the screen at column 40, using L$ (chr$(8)), which moves the cursor left:

```
x=cursor(0) : margin=40
while mod((x-1),80)+1 > margin
  print L$ : x=cursor(0)
endloop
var
  putcurs:integer;

begin
  page;
  {clears screen and homes cursor}
  putcurs := 45191;
  sysproc (putcurs,4618);
  writeln('Printing starts
  at row 18, column 10').
end.
```

You may temporarily intrude (with the TAB key, for example) beyond column 40, but you will be thrown out. With slightly more complex code you can create zones on screen which are totally forbidden and even force the cursor to skirt their borders.

Cursor Control, Other Languages

In all other SuperPET languages, you sense and move the cursor with two SYS functions: TGETCURS_ (which returns current location), and TPUTCURS_ (which moves the cursor where you want it). These two system functions employ the row as the high byte and the column as the low byte—modulo 256 (not modulo 80, as in mBASIC). It's not hard. I use "position" as a variable name for cursor location, and give "position" a value of 4383. (High byte is 43, low byte 83.) Grab a calculator and try this on 4383:

row = integer part (position/256) = 17
column = mod(position,256) = 31

In microFORTRAN or mPASCAL, if you define "position" as an integer, the code is very simple (APL works the same way, but I can't print the code here!):

row = position/256
column = mod(position,256)

If you want to pass the value picked up by TGETCURS_ to TPUTCURS_, you need not go through the conversion above. Both use the mod 256 values as is. But if you want to move the cursor to a new location, you have to know how to convert. Here is an example, in microPASCAL, in which I clear the screen and put the cursor on row 18, column ten (row = 18*256 = 4608 and column = 10. Final value is 4618). The decimal memory location of TPUTCURS_ is 45191, while that of TGETCURS_ is 45188 (the hex locations are $B087 and $B084, respectively).

You may, of course, write functions in all languages which convert to row and column so you needn't do the math yourself. TGETCURS_ and TPUTCURS_, if substituted for the cursor location techniques in microBASIC, work in exactly the same way in all other languages, though they take a bit more code.

Note that when we SYS, we first call the system procedure and then pass to it the desired parameter.

In microFORTRAN, you must employ a dummy argument (much as we did in mBASIC; in x= cursor(0), the "0" is a dummy argument). In addition, in the following, we use the hex address of TGETCURS_ to avoid converting to decimal. "Cnvh2i" is an intrinsic mFORTRAN function that converts the hex address to a decimal integer; "replace" is a variable that stores cursor position:

Sense cursor position:
replace = sys(cnvh2i('b084'))
Move cursor to that position:
y = sys(cnvh2i('b087'),replace)

The "y", above, is the dummy argument we men-

tioned. In APL, the SYS procedures work in the same way they do in mFORTRAN and in mPASCAL.

You now have the framework for absolute cursor control in all SuperPET languages.

Reviving Dear, Dead Programs

If you've ever said goodbye to a language and a long, tough program, only to find that you forgot to save it to disk or printer, you can revive the poor thing with a simple program called "retrieve," which was written by Waterloo Computing Systems at the University of Waterloo. It's in assembly language. No, don't run—you need know nothing about assembler to use it. If you can breathe, read and type, you can assemble and link it.

Return to the main menu and select the DEVELOPMENT package with: d <RETURN>. When you get a second menu, choose the microEDITOR with: e <RETURN>. In the mED, enter the program "retrieve.asm" exactly as it is printed in Listing 1. Make no changes!

When you've proofread it, file it on drive zero with: p retrieve.asm <RETURN>. Then clear the microEDITOR screen with: *d <RETURN> and enter "retrieve.cmd", again exactly as written. File it to drive zero with: p retrieve.cmd <RETURN>.

The hard work is done. Quit the microEDITOR with: bye <RETURN>, and you'll be back at the DEVELOPMENT menu. Choose the assembler with: a <RETURN>, and when the assembler asks for the filename, give it: retrieve <RETURN>. *Do not* put the ".asm" on the filename! The assembler will do all the rest of the work while you watch. When the assembler says it's through, hit <RETURN>, which takes you back to menu. Select the linker with: 1 <RETURN>, and it'll ask you for the filename to be linked. Give it: retrieve <RETURN>, with no ".cmd" on the filename. When the linker says it's through, hit <RETURN> to leave.

Then "quit" DEVELOPMENT and go back to main menu with: q <RETURN>, load the microEDITOR, and "get" your assembler file with: g retrieve.asm <RETURN>, because we're now going to "lose" that file and "retrieve" it.

Having assembled and linked "retrieve", you'll find a PRG file on the directory of drive zero called "retrieve. mod"; this is the machine language module which revives the dead. Let's see what it will do.

You're in the mED, with a file on screen, aren't you? Say: bye <RETURN> at command cursor. You have just lost a program and your file, and you're back at main menu. Retrieve your file and the microEDITOR with: disk.retrieve.mod <RETURN> and you will very quickly be back in the microEDITOR *with* your dear departed file. "Retrieve" will get any language back, with its program—so

long as you haven't switched to 6502 in the meantime. Don't use "retrieve" to return to DEVELOPMENT, which is menu drive; "retrieve" can't tell which program it's supposed to retrieve.

Last, copy "retrieve.mod" to your language disk as "retrieve". Call it from that disk at main menu any time with a simple: retrieve <RETURN>, and weep no more for that great program you killed.

Dick Barnes is editor of the SuperPET Gazette, *P.O. Box 411, Hatteras, NC 27943.*

Retriever

```
; retrieve.asm, written by Waterloo Computing Systems Ltd.

process equ     $2a
procalv equ     $39
curbnk_ equu    $220
bnkctl_ equ     $effc

retrieve equ     *
         ldd     #procalv ; get address of process auto-load vector
         std     process  ; point to it
         ldb     #$00     ; select bank 0
         tfr     cc,a     ; save condition code register
         sei              ; disable interrupts
         stb     curbnk_  ; remember new bank number
         stb     bnkctl_  ; set new bank number
         tfr     a,cc     ; restore condition code register
         ldd     #-1      ; NMI reset is TRUE
         std     2,s      ; replace old parameter value
         ldd     #1       ; INIT code
         jmp     $9000    ; leap to whatever is in Bank-Switched memory
         end
; Stop listing retrieve.asm on the line above. Do not copy this line.

; The next program is retrieve.cmd. Do not copy this line.

"retrieve"
org $680
"retrieve.b09"
; Do not copy this line or the lines below.
```

C

Using the Commodore B128's BLOAD and BSAVE Commands

by Al Fragola

The Commodore B128 has a pair of very powerful commands in BLOAD and BSAVE. They allow the saving and loading of designated blocks of memory to and from disks without resorting to a machine language monitor. Further, BLOAD allows you to locate the starting address of the binary code at any point in memory that suits your needs. It only takes a little imagination to see how this can be a powerful tool for the programmer.

The general formats for these two commands are as follows:

BLOAD"filename",B(bn),P(sta)
BSAVE"filename",B(bn),P(sta) TOP (fina)
Where bn=bank number, sta=starting address and fina=final address.

Using literals, the command would appear:
BLOAD"TEST",B2,P25 Which loads file "TEST" into bank two starting at address decimal 25. If the P element is left out, the start address will default to the start address used during the BSAVE.

Using variables, the command would appear:
BSAVE (F$),B(I),P(X) TOP(Y) Which saves the contents of bank one from decimal X to Y with a disk file name of F$.

To perform a SAVE WITH REPLACE, use "@TEST" or ("@"+F$) as the filename.
Putting this to work for us we can create a number of useful features. Let's look at several.

One-Line Program Append Command

Since program files can be BLOADED to any target starting address, it is possible to append one program to another with a one-line command. All that is done is to call the second program into bank one at a location which is two bytes below the "top of BASIC" pointer. We can get this location by incorporating two PEEKs into our BLOAD command:
BLOAD"pgmname",B1,(PEEK(47)−2+PEEK(48)*256)))
When you get a "READY", type in "0" and hit return. This adjusts all the pointers in BASIC to the new program size. Of course you should observe all the normal precautions for appending, such as:

- The appended program should start with a line number greater than the highest line number in the base program.
- Watch out for GOTO or GOSUB commands that will not have a valid destination address.
- Watch out for repeat DIM statements.

This simple routine can allow you to set up a series of regularly used routines on disk and append them as you build programs. The routine being called in for appending should have been stored on disk using the standard DSAVE type of protocol, not BSAVE.

Saving Variables Exactly as Stored in RAM

Have you ever entered a huge pile of data into your machine, only to find a fatal SYNTAX ERROR in your write-to-disk routine? Fear no more with the B128. Simply close the disk files and save the contents of the variable areas in bank two along with some of the bank 15 pointers. Then clean up your software sin and BLOAD the files back into the machine as if nothing happened. You will have every variable back and waiting for you to execute a GOTO to reenter the program. Just steer clear of DIM statements. It goes like this:

1. Close open disk files with DCLOSE

2. Save the variables, arrays and string pointers:
BSAVE"filname.1",B2,P(PEEK(49)+PEEK(50)*256) TOP (PEEK(57)+PEEK(58)*256)

3. Save the string images (only if there are any) by:
BSAVE"filname.2",B2,P(PEEK(59)+PEEK(60)*256) TOP (PEEK(63)+PEEK(64)*256)

4. And finally save the variable pointers:
BSAVE"filname.3",b15,P49 TOP 64

5. Now clean up the problem in the program and then:
BLOAD"filname.1",B2
BLOAD"filname.2",B2 (only if you saved strings)
BLOAD"filname.3",B15

6. Everything is like it was just before the BIG CRASH so just pick a safe reentry point and GOTO there.

By now some of you may have gotten the idea that this might be a faster way to save and load data files from disk. Since both BLOAD and BSAVE can be executed from BASIC, you are very correct. Every-

thing will be returned to the machine exactly as stored in memory. That includes the dimensions of any arrays as well as any temporary variables that were created. The savings in time will depend on how many strings are involved, as well as how many variables are in memory that normally wouldn't be stored using PRINT# statements.

The reason for the former is that strings are stored as pointers in lower RAM, with the images in upper RAM. This approach stores both, increasing file size. In a test using 1000 four-character strings in an array, the binary method was virtually the same speed, but the total disk space used was 30% greater. However, using an array of 1000 numeric values, both read and write times were reduced by 50%.

Whatever you choose to do in this area, be sure to have at least one string variable in memory so the bottom of string area pointer is brought down a couple of bytes from the top. To use this technique, you would simply use the above routines as part of your BASIC program. Prior to executing the BSAVE data save, place the statement PRINT FRE(2) to perform a quick garbage collection to reduce the string image area to minimum size. Also, if no strings have been created, it will be necessary to declare one (z$="xxxxx"+" ") to place something in upper RAM for storage.

Keep one point in mind, however. After a BLOAD data operation as described above, the only variables in memory will be the ones in the BLOADed data file. Further, DATA statements for strings and literals will not be saved.

Stored Screen Displays

The B128 allows you to define a "window" on the screen which is protected from material that is printed to the screen. Once these upper and lower boundaries are defined, however, it can be awkward to place new printing outside the window. Here's another perfect job for BLOAD and BSAVE. Simply prepare templates at the top and/or bottom parts of the screen outside the "window", and store and retrieve them from disk as binary files using these two commands. Once stored, they can be called in as part of your BASIC program to get the desired results. In fact, you can call in an entire screen if you desire. Let your imagination and creativity be your guide here.

Program Overlays

Although 64K of program RAM should be enough to satisfy most folks, one of the cardinal rules of computers is, "The program will always find a way to grow to a size slightly larger than available memory". Using BLOAD, it is possible to overlay one program over another in RAM starting at a specified line number. This is similar to the append operation above, except you target your overlay BLOAD to the point in RAM where the line link at the beginning of the program line in question starts. An array can be set up holding the various overlay file names, in the order you wish to have them called. Each must be called in to the same location in RAM, but this can be worked out easily. The overlays should have been saved using the regular DSAVE approach to avoid problems. BLOAD calls these in with ease. Program 1 gives a brief idea of how to do this.

The basic rules for using this technique are:

- The overlay(s) must be targeted to begin at a line that is later in the program than the BLOAD line.
- The line numbers in the overlay(s) should be higher than the highest line number in the base program.
- The last statement in the overlay should be a GOTO returning control to the base program if further overlays are to be called in. Line 30 in Program 1 would be the appropriate target.
- Literals and strings defined with DATA in the overlays will not be present properly in subsequent overlays unless you redefine them once after they are created.

It is easy to see that these two commands offer a wide variety of uses both within BASIC and as stand-alone commands. With a little bit of work, you should be able to find more and even better uses for BLOAD and BSAVE.

Program 1

```
10 REM the array f$(i) holds various overlay
   names
20 I=0
30 I=I+1
40 BLOAD(F$(I)),B1,P(t)
50 REM
60 REM
```

This is an oversimplification, but it should illustrate the point. The trick here is to determine the point in memory at which to start the overlay, which is the decimal address represented by the variable "t". Let us assume that we wish to target the overlay to begin over line 60. This can be done with the following direct command:

```
BANK1:FORX=2TOZ:IFPEEK(X+3)=60ANDPEEK
   (X+4)=0THENPRINTX+1:ELSENEXTX
```

For "Z" use a decimal address high enough to go through the program at least well past the desired line number. The two PEEKs look for the desired line number in the usual Commodore low/high format. The number that is printed to the screen will be the value of "t" to use. Be careful that you do not change the number of characters prior to line 60, or your target address will be wrong. One way would be to use "P00000" in line 40, then place the value of "t" in the line leaving the appropriate number of leading zeroes (e.g., P00189). C

Commodore's New MPS-801 Printer

Dot Matrix with Flair

by Richard Winters

I recently purchased the MPS-801 printer from Commodore. I had long awaited that day; here I was the proud owner of a Commodore 64, 1541 disk drive, CP/M™ cartridge, and 1600 modem, but no printer to access hard copy of my work. Having no previous experience with matrix printers, I was pleased with the ease of setting it up. The user's manual was loaded with excellent step-by-step illustrations and photos. I encountered no major complications and had my MPS-801 set up and ready to go in under 20 minutes. I spent the following 30 minutes scanning the user's manual to get some kind of grasp of things before attempting anything on my own. I strongly suggest that all those unfamiliar with printer operations thoroughly read the manual at least one time prior to experimentation to avoid any aggravation.

On power up, the MPS resets all of its internal switches to match the printing mode of the Commodore 64. Your new printer is capable of printing both capital and small letters, numbers, and all graphic characters available on the 64 in addition to user-definable printer graphics. The MPS is a computer in itself. It contains ROM, RAM, and a microprocessor. Therefore, it is a smart peripheral that will meet all of your printer output needs without reducing your computer's available memory. The printer buffer is only 90 bytes long, but Commodore guarantees that there will be no data loss due to overflow. This is possible because of the "handshaking" feature between Commodore peripherals and the computer system. Up to four disk drives and one more printer can be daisy-chained to the 64.

You can use the MPS in both the direct mode and under the control of a program. The commands associated with the MPS include OPEN, CLOSE, CMD, and PRINT#. They become easier to use as you continue using your printer. Here are some samples.

Example 1.
```
OPEN4, 4
CMD4
LIST
CLOSE4
```

Example 2.
```
OPEN15,4
PRINT#15,"COMMODORE"
PRINT#15,"MPS-801"
CLOSE15
```

Example 1 above would provide you with a hard copy listing of the program currently in memory. Example 2 would output the words "COMMODORE" and "MPS-801" to the printer.

There are an untold number of uses for a printer, some of which may include program listings, graphs, bar charts, statistical and mathematical curves and word processing.

There are also many diverse and useful printing modes accessible on the MPS-801, including standard character width for 80 columns, double character width for 40 columns, cursor-up mode for uppercase/graphics, cursor-down mode for uppercase/lowercase and graphic mode for defining graphics. All of these modes are accessible with the PRINT# and CHR$(X) commands of CBM BASIC. You can have a mixture of various printing modes activated at the same time as long as they are not contrasting.

Some additional features include reverse field off for black on white, reverse field on for white on black, linefeed, carriage returns, tab setting and dot addressing capability. Many of these modes are self-explanatory. The two that I found most interesting were the graphic mode and the dot addressing mode. The graphic mode allows you to define your own printer graphics within a 6×7 dot matrix. The dot addressing mode allows access to each of the 480 dots that a printing line consists of. With these two modes there are many possibilities, such as hi-res screen dumps, lo-res screen dumps, games with printing interaction and map drawing. The user's manual includes a very good screen copy program.

The MPS-801 printer comes in an attractive tan/brown casing. All the necessary cables are included along with a removable ribbon cartridge. Though the user's manual refers often to both the Commodore 64 and VIC 20, the boxes are labeled "C-64 ONLY".

[**Ed. note:** *Although the MPS-801 was originally planned to be compatible with both the VIC and 64, the final product is compatible with the 64 and the Commodore 264 series computers, and not the VIC. The first manuals, which were printed in advance, do not reflect the change. Later manuals should.*]

Commodore packed a lot into this package, with features such as an adjustable tractor mechanism,

removable printer cover, device selector switch and a removable ribbon cartridge with refillable inker. The device selector switch is one of the most exceptional features because it lets you make your MPS-801 printer either device number 4 or 5. Using this feature you can operate two MPS-801 printers without tying up the serial lines of your computer. There is also a T position for a self-test of the printer. The self-test prints out all available characters until changed or the printer is shut off. The printer cover reduces a fair amount of the noise produced while the printer is in operation.

If you shop around, you will find that the MPS-801 is priced comparably to the 1525 printer. The MPS-801 is about half as noisy as the 1525 and almost twice as fast, with a printing speed of 50 CPS (characters per second) compared to 30 CPS on the 1525. If you have any trouble with your new printer, Commodore has included in the user's manual a maintenance trouble shooting chart. If this is of no help, you can call Commodore Customer Support (215-436-4200) or Commodore Customer Service (215-431-9105).

To get you started with your new printer, I have included a short program to aid in user-defined printer graphics.

All in all, the MPS-801 printer incorporates speed, silence, and reliability into one sleek unit. My theory on the use of the letters "MPS" is that they stand for Matrix Printing Superiority. You will find that the MPS-801 has the Commodore performance that we have all come to expect.

Printer Character Editor

```
10 CLR
20 PRINT"[CLEAR]USE ONLY THE LEFT
   CURSOR KEY,"
30 PRINT"RIGHT CURSOR KEY,
    AND THE (*)"
40 PRINT"ASTERISK KEY DURING
   EXECUTION."
50 PRINT"USE THE RETURN AT END OF
   EACH"
60 PRINT"LINE.[SPACE2]
   TYPE DONE.. WHEN FINISHED"
70 PRINT"[SPACE12,SHFT E6]"
80 PRINT"PRESS ANY KEY TO BEGIN"
90 GET A$
100 IF A$=""THEN GOTO 90
110 DIM A$(255)
120 LET D=0
130 DIM C(5)
140 IF D=256 THEN GOTO 480
150 FOR F=0 TO 5:LET C(F)=0:NEXT F
160 PRINT"[CLEAR]";
170 FOR F=1 TO 7:PRINT"
    [SPACE2]......":NEXT F:PRINT
    "[HOME]";
180 FOR F=1 TO 7
190 INPUT C$
200 IF C$="DONE.."THEN GOTO 480
210 NEXT F
220 FOR F=0 TO 6
230 FOR G=0 TO 5
240 IF PEEK(1026+G+(F)*40)=42 THEN
    LET C(G)=C(G)+2^F
250 NEXT G
260 NEXT F
270 A=0
280 A$=""
290 FOR F=0 TO 5
300 LET A=C(F)+128
310 A$=A$+CHR$(A)
320 NEXT F
330 OPEN 4,4
340 FOR I=1 TO 10
350 PRINT#4,CHR$(8)A$;
360 PRINT#4,CHR$(15)" ";
370 NEXT I
380 PRINT#4
390 PRINT#4,CHR$(15)"....DATA
    IS ";
400 FOR F=0 TO 5
410 LET Z=C(F)+128
420 PRINT#4,Z",";
430 NEXT F
440 PRINT#4
450 LET D=D+1
460 CLOSE 4
470 GOTO 140
480 PRINT#4
490 CLOSE 4
500 END
```

C

Commodore's *Easy Spell* for the 64 Revisited

by Francis V. Amato

For about $20.00 you can avoid misspelled words forever.

I'm a phonetic speller. That's another way of saying I can't spell. If you think the word "Commadore" looks right, have I got a piece of software for you! It's easy to use, works great and is cheap! What more could a poor speller ask for?

Word processing is increasingly important in the modern world. Effective communication is quite literally the very foundation of success. Most of us need to type letters. A few of us make our livings manipulating words. If you do anything more than the most elementary writing, you need a spelling checker. I use Commodore's *Easy Spell* with their *Easy Script* word processor. *Easy Spell* makes a good word processor even better. Not only will it correct my mistakes, but since using it I'm learning to make fewer of them.

Easy Spell requires a Commodore 64, the *Easy Script* word processor from Commodore and one or more 1541 disk drives (or a dual drive with IEEE interface). The program comes in the standard Commodore folder. There's an *Easy Spell* program disk, which cannot be copied, and a dictionary disk which can. A postcard's included for a five dollar replacement if the master disk is damaged. That's more than reasonable when you consider *Easy Spell* costs only about twenty dollars!

The dictionary disk comes with 20,000 words. According to the manual, there are several optional dictionaries available. You can buy an extended version with 32,000 words; a UK dictionary (aeroplane instead of airplane); a combined UK/US version or a pre-formatted blank master dictionary to create your own lists of words.

In addition to the master dictionary, *Easy Spell* allows you to create your own list of words. The user dictionary is searched along with the master. Some spelling checkers don't do this automatically but make you specify the dictionary to search, which can be extremely inconvenient.

You load *Easy Spell* from within *Easy Script*. After the title screen, it asks if you're using a single (default value) or dual drive. The option screen offers seven choices. You can check an *Easy Script* file, get a word frequency report, print the user dictionary, delete from the user dictionary, search the dictionaries, change printer specifications or end the program.

Check an *Easy Script* text file: This is a default response requiring only a press of the return key. It prompts you to remove the *Easy Spell* disk and insert the text disk. It reads the disk directory and asks for the name of the file or allows you to use the F1/F3 keys to scroll through the directory. After choosing, you're asked if it's a linked file. If yes, it will read in and check each part. The manual makes no mention of the size file allowed. I regularly check 4000 words and have no problems. You needn't fear if the file is too large. It will accept as much as possible then allow you to check the remainder.

Easy Spell reads the file. When it's finished, you insert the dictionary disk. It takes about eight minutes to read and check a 4000 word document. Rather slow, but more than adequate for nearly all users. It tells you the total number of words in the file, the number of unique words, the number of sentences and paragraphs and the average word length. Finally, it proceeds to list the words it doesn't know. If you want the list printed, it will be. If you want the file edited it prompts you to reinsert the text disk.

The program then shows you a full page of text with any unknown words highlighted in reverse video. You can correct, ignore or add them to the auxiliary dictionary. After checking all the words, *Easy Spell* replaces the original file with the edited copy. If you had chosen to add to the dictionary, you would reinsert the dictionary disk. It's back to the *Easy Spell* menu when finished.

Get a word frequency report: I didn't believe this option would be of much use to me. I was quite mistaken. It reads the text file and outputs (to the screen or printer) a list of the unique words and how many times each was used. This can be a real eye opener. Like many people, I use certain words far too often. This option allows you to check so you can go back and replace them with synonyms or rephrase your sentences. Boring text is unread text. This feature is also useful for educational purposes,

because you can see if certain words are inappropriate for the group you're writing for. So I have come to appreciate it.

Print and/or delete from the user dictionary: You should periodically review your dictionary. Adding much more than a thousand words to the user dictionary slows the program down considerably. I often use this feature. Experts claim the majority of the population can function adequately with a vocabulary of no more than 3000 words. You'll soon discover you have a basic vocabulary and a few words which change depending on what you're working on. Unnecessary words are just excess baggage. You can delete entire groups of words with wild card characters.

Search dictionaries: You can search either the master or user dictionary with or without wild cards.

Change printer specs: Lets you change the printer specifications (i.e., printer type, width, etc.). I rarely use it, but those times I have, it sure came in handy. Otherwise, I would have to abort the program, go back to *Easy Script,* change things and reload *Easy Spell.* If you need this option even once in your life, you'll be happy you have it.

End program: This returns you to the main option screen of *Easy Script.* Any file in memory before loading *Easy Spell* will be erased so be sure to save it.

Other than its slowness in checking a text file, I have absolutely no complaints about *Easy Spell.* I tried hard to find some but was unsuccessful. It's copy-protected, which I normally don't like, but receiving two copies and a card for replacement makes up for it. It can only be loaded from within *Easy Script,* which is slightly annoying, but I couldn't honestly call it a complaint.

It's true there are certain features that a few (very few) spelling checkers offer that are not contained in *Easy Spell.* Features like a built-in thesaurus, the ability to co-reside with the word processor, the option of checking spelling while creating text. But these programs cost hundreds of dollars, and the extra features aren't really worth it unless you have unusual needs.

One of the great features of *Easy Spell* is that mistakes are shown in context. Many spelling checkers don't offer that, but give you just the word itself, or only the line it's in, or make you use a search option to find the mistake. These ways are tedious and potentially dangerous.

Simply put, *Easy Spell* is outstanding. I can't believe it costs a mere $20. It's really an incredible bargain. If you have *Easy Script* and don't have *Easy Spell* you're missing an important program.

I can't tell you how much I recommend *Easy Spell.* I seriously wonder how I ever did without it. C

Re-Invention of the Paperclip

by Kelley M. Essoe

The Original:

pa'per-clip (pa'per-klip),n.: A flexible tool for holding loose scraps of paper together.

No surprises here. We all know that the paperclip is a remarkable invention both in concept and simplicity of design. It is functional, versatile, easy to use, reasonably priced and does exactly what it was intended to do... and more. (Hand one to any kid for a demonstration of its less obvious potential uses.)

The New Original:

PaperClip™ (pa'per-klip),n.: A flexible tool for holding (and creating, saving, merging, moving, deleting, inserting, copying, transferring, appending) loose scraps of ideas, thoughts, notes, novels, scripts, stories, lists and letters together.

Surprise! The newly designed *PaperClip* is also functional, versatile, easy to use, reasonably priced, and does exactly what it was intended to do... and more.

Move over *WordPro.* Not so fast *Quick Brown Fox.* A Canadian-based software company with a tongue-in-cheek name has a power-packed word processor for the Commodore 64 computer. The *PaperClip 64C* word processor from Batteries Included has arrived.

PaperClip's capabilities are impressive. With well over 150 different control, formatting and operation commands, just about every feature you could possibly want or need is here. And that includes the all important but oft disregarded quality of ease of use.

Getting Started

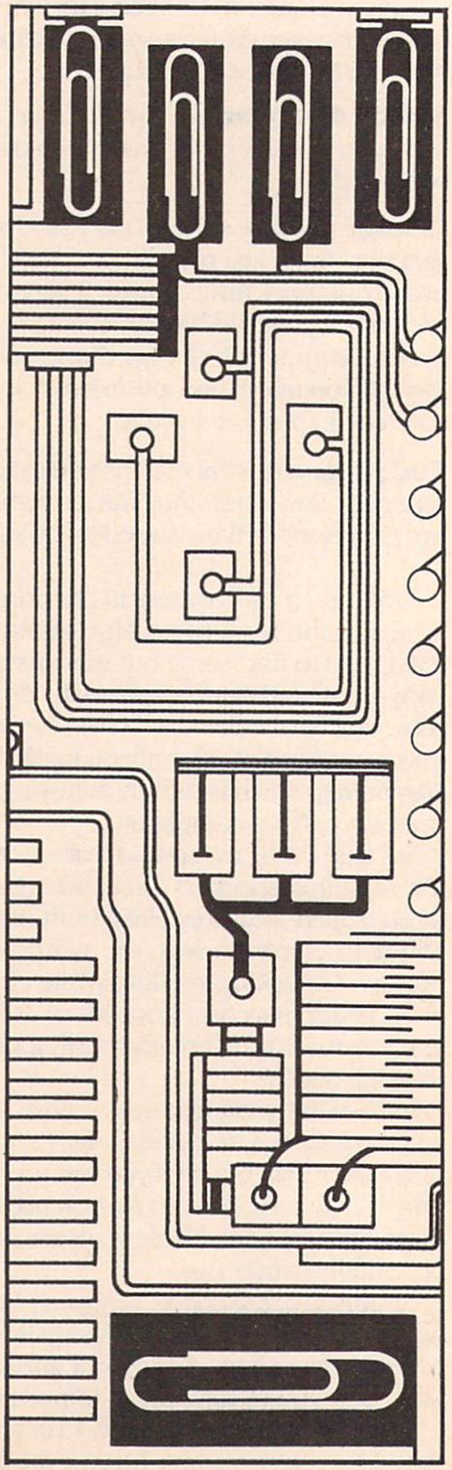
Unlike many other word processing programs for the 64, *Paper Clip 64* comes furnished on an unprotected disk. This means you can create as many back-up copies as you need in order to feel that your investment *is* protected. The *PaperClip* disk even includes a single drive back-up program for this purpose. Batteries Included safeguards itself against unauthorized copying by the use of a dongle, or electronic key, which fits into port number one. *PaperClip* won't run without it.

When the program runs, you will notice the top two lines of your screen are separated from the rest of your work area. These are the tab line and the status line, respectively. Tab stops, prompts, current cursor position and various status indicators are displayed on these "reserved" lines.

If you are using a color monitor, such as the Commodore 1701 or 1702, *PaperClip* allows you to change the background, border and cursor colors at any time with the F2, F4 and F6 function keys. This is a handy little feature that allows you to find your own ideal visual combination.

In order to start writing... you start writing. When you are typing along and reach the end of a line, simply keep typing. Text will wrap around automatically to the beginning of the next line.

A discouraging drawback to word processing on the Commodore 64 is its 40-column screen display limitation. Writing on a 40-column screen is different from working with the standard 80 col-



umns in a number of ways. The major difference is that what you see is *not* what you get. It is important therefore that when working with 40 columns that you do not edit your text so that it looks all nice and neat on the screen. It will end up looking all strange and funny on paper. However, *PaperClip* takes this basic Commodore 64 inconvenience and provides the user with a number of compensatory options, as you will soon discover.

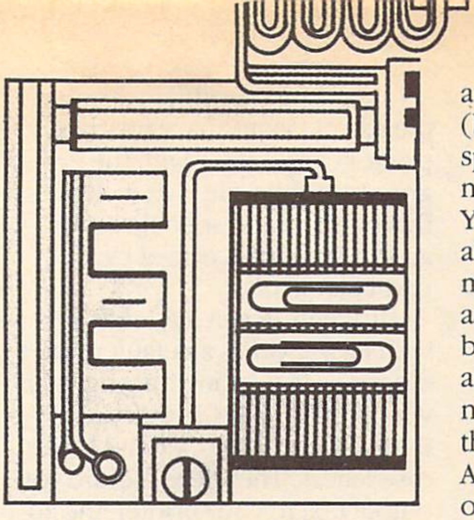
Text Editing

You want the ability to draft, revise, rearrange and polish your text quickly and easily? You got it.

First of all, *PaperClip*'s editor is fully screen-oriented. You can move the cursor in all four directions. You can scroll... up to the beginning and down to the ending of text. You can even change the line length and scroll sideways *up to 250 columns across* (which is one of the best ways to get around the 40-column aggravation). An accelerated scroll function lets you race through text at high speed, which is quite useful for moving quickly to a certain part of a long document.

So all you have to do is move the cursor to the place in the text you want to change and type in the change. Easy. Type over errors. Delete a letter, word, phrase or paragraph. Insert a couple of words or a couple of pages and watch the following text move over to make room.

But what if you habitually put "i" before "e" even if it comes after "c"? Or maybe you discover at the end of a twenty-page presentation to Mr. R.D. Schmitt that you have consistently misnomered him R.D. Schmutt? *PaperClip*'s "search and replace" function can easily rectify the problem. Type in the "search" text; type in the "replace" text. *PaperClip* will replace each occurrence of the search string with the replace string. You even have the option to approve replacements one by one as each individual



search string is encountered. *Voila!* Mr. Schmitt will never know.

The search and the search-and-replace functions can be utilized both locally and globally, and *PaperClip* supports extensive "wild card" searches as well.

Other invaluable editing tools furnished by *PaperClip* are the "cut and paste" and "pre-defined phrase" functions.

The former allows you to "set", or define, a block of text, which you can then transfer or copy to another place in the document, save to another file or delete completely.

The latter gives you the ability to define a single key as any one-screen-line phrase. For example, say you are writing a legal document in which you will frequently be referring to various persons in "legal-ese". You can define the key "g" to equal the phrase "Glen Good, the party of the first part" and the key "b" as "Brad Bad, the party of the second part". Now, instead of having to type out Glen's or Brad's extended titles simply type <esc> "g" or <esc> "b" and let *PaperClip*'s fingers do the walking.

Text Formatting

From form letters and mailings to screenplays and manuscripts, from business proposals and Dear John letters to scientific dissertations and mathematical tables and columns, *PaperClip* gives you over 30 commands to meet just about every imaginable formatting requirement. For instance, margin settings, page length, paging, automatic indentation, tab stops (both regular and numeric), line spacing, headers, footers and page numbering are all user-defined. You can center, justify and right-align your text as you please, and mix and match print types such as pitch settings, underlining, boldface, italics, superscripts and subscripts. Plus, there are nine user-definable characters that you can use to send down ASCII values to the printer in order to print special characters that are not normally available through the word processor.

Other useful features include the ability to automatically generate a table of contents, add non-printing comments within text and use a pause command that causes the output to be stopped temporarily and a user-defined message to be printed on the top line of the screen such as, "Change print wheel now," or "Have you eaten yet today?"

PaperClip also has a number of advanced features not commonly found in basic word processors. One very powerful function is its ability to manipulate columns. You can set, move, repeat, shift and delete them. You can perform simple mathematical calculations on them such as totaling a column of numerical data. You can even have the contents of a column sorted into ascending or descending order. For example, you can alphabetize the names and addresses in a mailing list or sort numbers according to value.

Output

So, you have finished writing, editing and formatting your document. Now you'd like to preview its general appearance.

As previously pointed out, your work/edit/formatting screen is not going to look like your end product. In this "working" mode you will have many format commands, arrows, checkmarks and such interspersed throughout your text, making it less than inviting for a good read through. To view your

text without the "working" commands, *PaperClip* gives you three output options. Two of these are video-directed and the third is, of course, printed hard copy.

The first of these is the "normal" video display, formatted to 80 columns that are viewed by scrolling left and right. The Commodore 64, as we've already mentioned, is limited to a 40-column display. So looking at its normal video representation of a standard 80-column sheet is a bit like looking at a panorama with blinders on. You have to take it in an eyeful at a time, or, in this case, 40 columns at a time, so it's difficult to get a decent preview of the final page.

If you need to view text on the screen just as it will appear on the page, *PaperClip* comes to the rescue to prevent the migraines that result when you try to size up a pachyderm through a peephole. Without additional hardware or interface cards, you can also view an *80-column video display* of your document. *PaperClip* achieves this through the use of high-res graphic characters that are half as wide as the normal characters. However, the 80-column display is for viewing only. It is not available in the editing mode, because you are not looking at normal text, but rather a graphics display that *emulates* the characters. Nevertheless the 80-column display is a real help when you want to get a realistic preview of a finished page before you print it out.

One of the stumbling blocks to producing printed output from a word processor is that the actual code sequence to turn on and off features such as backspacing, underlining, boldface, etc., *varies* from printer to printer. It is because of this basic lack of standards that the large majority of word processors tend to support a limited number of printers or require dealership intervention to permanently configure specific printer codes into the word processing program.

PaperClip implements a unique solution to this problem. Through the use of printer files, nearly any printer can be used with the program. A printer file is a small file that contains the specific codes and code sequence used by a particular printer.

Although *PaperClip* loads in with PET ASCII as a default printer file, you can't do anything fancy with it. To PET ASCII, fancy is anything other than upper and lower case letters. Therefore, before you output text to your printer, the appropriate printer file must be loaded from within *PaperClip* first.

At the time of this review, *PaperClip* comes with printer file support for 24 of the most popular printers. If by some unlikely event you find that your printer is not among them, there is a program on the *PaperClip* disk called "Printer Set-up", which will allow you to create your own printer file. Now *PaperClip* will support your printer, too, so you can load in your printer file at the start of each new session or make use of yet another program on the *PaperClip* disk, which allows you to easily configure your specific printer file into *PaperClip* as the new default.

Printer output can be continuous or discontinuous, allowing for both fanfold and single sheet paper; it can be "global", so you can chain one file after another for output; and it can access sequential files to fill variable blocks such as those used in form letters. You can also switch between video output and printer output, begin output at any specified page of your document, automatically restart the previous output at the top of the last page that was printed or displayed or send the output to a disk file rather than to the printer.

More Mentionables

PaperClip also supports the ability to print in French or multilingual characters. These character sets are supplied on the *PaperClip* diskette and are loaded from within *PaperClip*.

At any time, and without overwriting any text currently in memory, it is possible to call up a directory of the files stored on the disk drive. The screen will blank, and each entry will scroll onto the screen in a three-component format: file name, file type, and the number of blocks used by the file. Hitting the CLR/HOME key twice takes you back to your text in the edit mode. You can also use the directory mode to load files into *PaperClip* using a handy "screen-read" function.

Other commands provided by *PaperClip* permit you to set or change device numbers, send direct disk commands, fetch disk status and load a disk directory directly into memory in order to edit, sort or print it out.

One Unmentionable

If there were anything about this word processor I would change, it would have to be the manual. Unfortunately someone forgot to include an index. Someone also forgot to organize the myriad of feature directives into a logical sequence. Particularly frustrating is the way that many of the features are treated with a cryptic "oh-by-the-way" approach. This is a powerful program that really ought to be paired up with equally superior documentation.

Fortunately, Batteries Included is working on a revision of the book, which will include an index, much more detailed instructions and a more organized format.

But, in general, the extensive scope of capabilities and the inherent flexibility of this program takes *PaperClip* out of the basic home word processing class and into the realm of the big guys. However, unlike some of the big leaguers, it is amazingly easy to use, won't make your bank account look like an anorexic's famine fund and runs on the best of the lower priced computers around. The paperclip sure has changed, hasn't it? **C**

Circle Reader Service No. 501

User Group Listing

ALABAMA

Huntsville PET Users Club
9002 Berclair Road
Huntsville, AL 35802
Contact: Hal Carey
Meetings: every 2nd Thursday

Riverchase Commodore Users Group
617 Grove St.
Birmingham, AL 35209
(205) 988-1078
Ken Browning

Wiregrass Micro-Computer Society
Commodore SIG
109 Key Bend Rd.
Enterprise, AL 36330
(205) 347-7564
Bill Brown

Tiger Byte: E. Alabama CBM 64 Users Group
c/o The Computer Store, Inc.
Midway Plaza
Opelika, AL 36801
Jack Parsons
1st & 3rd Wed. of Month

The Birmingham Commodore Computer Club
Birmingham, AL
(205) 923-9260
Harry Jones

Commodore Club of Mobile
3868-H Rue Maison
Mobile, AL 36608
(205) 343-1178
Tom Wyatt
3rd Thurs. of month

Shoals Commodore Users Group (SCUG)
209 Lakeshore Dr.
Muscle Shoals, AL 35661
Geo. Taylor
2nd & 4th Tues. of month

1920-A Avenue C
Brookly
Mobile, AL 36615
(205) 661-1973
Howard Crider

CC & Me
P.O. Box 324
Pinson, AL 35126
(205) 854-0650
Bill Freeman

1734 S. Atmore Ave.
Whistler, AL 36612
(205) 452-9740
William Autry

ALASKA

COMPOOH-T
c/o Box 118
Old Harbor, AK 99643
(907) 286-2213

Alaska 84 Computer Club
c/o Line 49 Management
P.O. Box 6043
Anchorage, AK 99502

First City Users Group
P.O. Box 6692
Ketchikan, AK 99901
(907) 225-5695
James Llanos

ARIZONA

VIC Users Group
2612 E. Covina
Mesa, AZ 85203
Contact: Paul Muffuletto

Catalina Commodore Computer Club
2012 Avenida Guillermo
Tucson, AZ 85710
(602) 296-6766
George Pope
1st Tues. 7:30 p.m.

Central Arizona PET People
842 W. Calle del Norte
Chandler, AZ 85224
(602) 899-3622
Roy Schahrer

ACUG
c/o Home Computer Service
2028 W. Camelback Rd.
Phoenix, AZ 85015
(602) 249-1186
Dan Deacon
First Wed. of month

West Mesa VIC
2351 S. Standage
Mesa, AZ 85202
Kenneth S. Epstein

Arizona VIC 20-64 Users Club
232 W. 9th Place North
Mesa, AZ 85201
Donald Kipp

Arizona VIC & 64 Users
904 W. Marlboro Circle
Chandler, AZ 85224
(602) 963-6149
Tom Monson

Canyon De Chelly—Four Corners Users Group
c/o Calumet Consulting
Box 1945
Chinle, AZ 86503
(602) 674-3421
Larry DiLucchio
they meet bi-monthly

ARKANSAS

Commodore/PET Users Club
Conway Middle School
Davis Street
Conway, AR 72032
Contact: Geneva Bowlin

Booneville 64 Club
c/o A. R. Hederich
Elementary School
401 W. 5th St.
Booneville, AR 72927
Mary Taff

The Siloam Commodore Computer Club
P.O. Box 88
Siloam Springs, AR 72761
(501) 524-5624
Ken Emanualson

Russellville Commodore User Group
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

Arkansas River Valley Commodore Users
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

P.I.C. Club
c/o Hatfield Public Schools
Box 130
Hatfield, AR 71945
(501) 389-6164
Bob Reed

CALIFORNIA

SCPUG Southern California PET Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: First Tuesday of each month

California VIC Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: Second Tues. of each month

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, Secretary

PALS (Pets Around) Livermore Society
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
7615 Leviston Ave.
El Cerrito, CA 94530
(415) 527-9286
Bill MacCracken

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 a.m.-4 p.m.

Walnut Creek PET Users Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

Jurupa Wizards
8700 Galena St.
Riverside, CA 92509
781-1731
Walter J. Scott

The Commodore Connection
2301 Mission St.
Santa Cruz, CA 95060
(408) 425-8054
Bud Massey

San Fernando Valley Commodore Users Group
21208 Nashville
Chatsworth, CA 91311
(213) 709-4736
Tom Lynch
2nd Wed. 7:30

VACUUM
277 E. 10th Ave.
Chico, CA 95926
(916) 891-8085
Mike Casella
2nd Monday of month

South Bay Commodore Users Group
1402 W. 218th St.
Torrance, CA 90501
Contact: Earl Evans

Slo VIC 20/64 Computer Club
1766 9th St.
Los Osos, CA

The Diamond Bar R.O.P. Users Club
2644 Amelgado
Haciendo Hgts., CA 91745
(213) 333-2645
Don McIntosh

Commodore Interest Association
c/o Computer Data
14660 La Paz Dr.
Victorville, CA 92392
Mark Finley

Fairfield VIC 20 Club
1336 McKinley St.
Fairfield, CA 94533
(707) 427-0143
Al Brewer
1st & 3rd Tues. at 7 p.m.

Computer Barn Computer Club
319 Main St.
Suite #2
Salinas, CA 93901
757-0788
S. Mark Vanderbilt

Humboldt Commodore Group
P.O. Box 570
Arcata, CA 95521
R. Turner

Napa Valley Commodore Computer Club
c/o Liberty Computerware
2680 Jefferson St.
Napa, CA 94558
(707) 252-6281
Mick Winter
1st & 3rd Mon. of month

S.D. East County C-64 User Group
6353 Lake Apopka Place
San Diego, CA 92119
(619) 698-7814
Linda Schwartz

Commodore Users Group
4237 Pulmeria Ct.
Santa Maria, CA 93455
(805) 937-4174
Gilbert Vela

Bay Area Home Computer Asso.
1332 Pine St.
Walnut Creek, CA 94598
(415) 932-5447
Cliff Downing

Amateurs and Artesians Computing
P.O. Box 682
Cobb, CA 95426

Manteca VIC 20 Users
Organization
429 N. Main St.
Manteca, CA 95336
Gene Rong

Pomona Valley Commodore
Users Group
1401 W. 9th, #77
Pomona, CA 91766
(714) 620-8889
Mark Joerger
1st & 3rd Wed. of month 7 p.m.

VIC TORII-The VIC 20
Users Group
PSC #1, Box 23467
APO San Francisco, CA 96230
Wesley Clark

The Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA 91506
1st Wed. 7 p.m.

The Commodore Tech. Users
of Orange Co.
P.O. Box 1497
Costa Mesa, CA 92626
(714) 731-5195
Roger Fisher

VIC 20 Software
Exchange Club
10530 Sky Circle
Grass Valley, CA 95945
Daniel Upton

C-64 West Orange County
Users Group
P.O. Box 1457
Huntington Beach, CA 92647
(714) 842-4484
Philip Putman
2nd & 4th Tues. of month

Antelope Valley Commodore
Users Group
POB 4436
Lancaster, CA 93539
(805) 942-2626
James Haner
1st Saturday

Diablo Valley Commodore
Users Group
762 Ruth Dr.
Pleasant Hill, CA 94523
(415) 671-0145
Ben Braver
2nd & 4th Thurs. 7:30 p.m.

CA. Area Commodore
Terminal Users Society
C.A.C.T.U.S.
P.O. Box 1277
Alta Loma, CA 91701
Darrell Hall

20/64
P.O. Box 18473
San Jose, CA 95158
(408) 978-0546
1st Sun. of month (6-9 p.m.)

8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Software 64
353 California Dr.
Burlingame, CA 94010
(415) 340-7115
Mario Abad

Sacramento Commodore
Users Group
8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Peninsula Commodore
Users Group
549 Old County Rd.
San Carlos, CA 94070
(415) 593-7697
Timothy Very
2nd Thurs. of Month

San Francisco Commodore
Users Group
278-27th Ave. #103
San Francisco, CA 94121
(415) 387-0225
Roger Tierce

South Bay Commodore 64
Users Group
P.O. Box 3193
San Ysidro, CA 95073

Commodore 64 West
Computer Club
2917 Colorado Ave.
Santa Monica, CA 90404
(213) 828-9308
Don Campbell

Sixty Fourum
P.O. Box 16098
Fresno, CA 93755
John Damiano

C-64/VIC 20 Users Group
Pasadena City College
Cicadian Room
Pasadena, CA
(714) 593-4880
Chuck Cypher
7 p.m. 1st & 3rd Thursdays

Marin Commodore
Computer Club
620 Del Ganado Rd.
San Rafael, CA
(415) 479-0426
2nd Wed. of month 7:30 p.m.

Santa Rosa Commodore 64
Users Group
333 East Robles Ave.
Santa Rosa, CA 95407
(707) 584-7009
Garry Palmer
meets once a month

The Exchange
P.O. Box 9189
Long Beach, CA 90810
(213) 595-1771
Michael C. Joseph, MD

Southern California Edison
Commodore Club
P.O. Box 800
Rosemead, CA 91770
Jerry Van Norton

LOGIKS Commodore
Computer Club
c/o Christ Presbyterian Church
620 Del Ganado Rd.
San Rafael, CA 94903
(415) 479-0426
Elmer Johnson
2nd Wed. of month 7:30 pm

Commodore 64 West
P.O. Box 346
Culver City, CA 90232
(213) 398-0913
Charles P. Santos

Commodore Users Group of
Riverside (CUGR)
P.O. Box 8748
Riverside, CA 92515
(714) 689-1452
Ken Brown
2nd & 4th Thurs. night

VIC-Club: San Francisco (VCSF)
1503A Dolores
San Francisco, CA 94110
Colin Johnston
ages 10 to 16 preferred.

SIG (Special Interest Group)
1135 Coronet Ave.
Pasadena, CA 91107
Brian R. Klotz

VIC 20 Software Exchange
7660 Western Ave.
Buena Park, CA 90620
Vincent Beltz

COLORADO

VICKIMPET Users Group
4 Waring Lane,
Greenwood Village
Littleton, CO 80121
Contact: Louis Roehrs

Colorado Commodore
Computer Club
2187 S. Golden Ct.
Denver, CO 80227
986-0577
Jack Moss
Meet: 2nd Wed.

Commodore Users Group
Box 377 Aspen, CO 81612
(303) 925-5604
Ray Brooks
1st Monday in the evenings

Vicdore Users Group
326 Emery Dr.
Longmont, CO 80501
(303) 772-2821
Wayne Sundstrom

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

Commodore User Club
Wethersfield High School
411 Wolcott Hill Road
Wethersfield, CT 06109
Contact: Daniel G. Spaneas

VIC Users Club
c/o Edward Barszczewski
22 Tunxis Road
West Hartford, CT 06107

New London County
Commodore Club
Doolittle Road
Preston, CT 06360
Contact: Dr. Walter Doolittle

The Commodore East
Users Group
165 B S. Bigelow Rd.
Hampton, CT 06247
(203) 455-0108

Commodore Users Group
of Stratford
P.O. Box 1213
Stratford, CT 06497
(203) 377-8373
Dan Kern-Ekins

Capitol Region Commodore
Computer Club
57 Carter Dr.
Tolland, CT 06084
Prudence Schifley
2nd Mon. of month 7 p.m.

CT Computer Society
180 Bloomfield Ave.
Hartford, CT 06105
(203) 233-3373
Harry Hill
last Sat. of month

Fairfield County Commodore
Users Group
P.O. Box 212
Danbury, CT 06810
Linda Retter

DELAWARE

The Diamond State
Users Group
Box 892, RD 2
Felton, DE 19943
(302) 284-4495
Michael Butler

Newark Commodore Users
Group (NCUG)
210 Durso Dr.
Newark, DE 19711
(302) 737-4686
Bob Black
once a month Newark H.S.

DISTRICT OF COLUMBIA

USO Computer Club
USO Outreach Center
207 Beyer Rd., SW
Washington, DC 20332
Steven Guenther

FLORIDA

Jacksonville Area
PET Society
401 Monument Road, #177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida
PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

PETs and Friends
129 NE 44 St.
Miami, FL 33137
Richard Plumer

Sun Coast VICs
P.O. Box 1042
Indian Rocks Beach, FL
33535
Mark Weddell

Bay Commodore
Users Group
c/o Gulf Coast Computer
Exchange
241 N. Tyndall Pkwy.
P.O. Box 6215
Panama City, FL 32401
(904) 785-6441
Richard Scofield

Gainesville Commodore
Users Club
3604-20A SW 31st Dr.
Gainesville, FL 32608
Louis Wallace

Brandon Users Group
108 Anglewood Dr.
Brandon, FL 33511
(813) 685-5138
Paul Daugherty

Brandon Commodore
Users Group
414 E. Lumsden Rd.
Brandon, FL 33511

Gainesville Commodore
Users Group
Santa Fe Community College
Gainesville, FL 32602
James E. Birdsell

Commodore Computer Club
P.O. Box 21138
St. Petersburg, FL 33742
(813) 522-2547
Chuck Fechko

Commodore Users Group
545 E. Park Ave.
Apt. #2
Tallahassee, FL 32301
(904) 224-6286
Jim Neill

The Commodore Connection
P.O. Box 6684
West Palm Beach, FL 33405

El Shift OH
P.O. Box 548
Cocoa, FL 32922
Mike Schnoke
Sat. mornings/every 4 to
6 weeks

Miami 20/64
12911 S.W. 49th St.
Miami, FL 33175
(305) 226-1185

Tampa Bay Commodore
Computer Club
10208 N. 30th St.
Tampa, FL 33612
(813) 977-0877

Commodore Computer Club
P.O. Box 9726
Jacksonville, FL 32208
(904) 764-5457
David Phillips
2nd & 4th Tues. of Month

VIC/64 Heartland Users Group
1220 Bartow Rd. #23
Lakeland, FL 33801
(813) 666-2132
Tom Keough
4th Wed. of Month at PRC

64 Educators Users Group
South
FDLRS-South
9220 S.W. 52nd Terrace
Miami, FL 33165
(305) 274-3501
Dr. Eydie Sloane

64 Educators Users Group
North
16330 N.E. 2nd Ave.
North Miami Beach, FL 33162
(305) 944-5548
Robert Figueroa

Suncoast 64S
c/o Little Professor
Book Center
2395 U.S. 19 North
Palm Harbor, FL 33563
(813) 785-1036
Curtis Miller

Lakeland VIC 20 Users Group
2450 Shady Acres Dr.
Mulberry, FL 33860

Broward Commodore
Users Group
13 Spinning Wheel Lane
Tamarac, FL 33319
(305) 726-4390
Lewis Horn

Volusia Ct. Commodore
Program Exchange
1612 Reynolds Rd.

DeLeon Springs, FL 32028
Rick Stidham

The Ultimate 64 Experience
5740 S.W. 56th Terrace
Miami, FL 33143
Sandy Cueto

Clearwater Commodore Club
1532 Lemon St.
Clearwater, FL 33516
(813) 442-0770
Gary Gould

The Commodore Advantage
P.O. Box 18490
Pensacola, FL 32523
(904) 456-6554
Deanna Owens
2nd Friday of month

GEORGIA

VIC Educators Users Group
Cherokee County Schools
110 Academy St.
Canton, GA 30114
Dr. Al Evans

Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

VIC-tims
P.O. Box 467052
Atlanta, GA 30346
(404) 922-7088
Eric Ellison

Golden Isles Commodore
Users Club
Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

Commodore Club of Augusta
1011 River Ridge Rd.
Apt. #14-A
Augusta, GA 30909
David Dumas

Atlanta Commodore 64
Users Group
1767 Big Valley Lane
Stone Mountain, GA 30083
(404) 981-4253
Ron Lisoski

Atlanta 64 Users Group
P.O. Box 5322
Atlanta, GA 30307
Phil J. Autrey

Albany Commodore Amateur
Computerist
P.O. Box 5461
Albany, GA 31706
David Via

HAWAII

Commodore Users Group
of Honolulu
c/o PSH
824 Bannister St.
Honolulu, HI
(808) 848-2088
3rd Fri. every month

20/64 Hawaii
P.O. Box 966
Kailua, HI 96734
Wes Goodpaster

Commodore Users Group
of Honolulu
1626 Wilder #701
Honolulu, HI 96822
(808) 848-2088
Jay Calvin (808) 944-9380

IDAHO

GHS Computer Club
c/o Grangeville High School
910 S. D St.
Grangeville, ID 83530
Don Kissinger

S.R.H.S. Computer Club
c/o Salmon River H.S.
Riggins, ID 83549
Barney Foster

Commodore Users
548 E. Center
Pocatello, ID 83201
(208) 233-0670
Leroy Jones

Eagle Rock Commodore
Users Group
900 S. Emerson
Idaho Falls, ID 83401
Nancy J. Picker

64 Bug (Boise Users Group)
P.O. Box 276
Boise, ID 83701
(208) 344-6302
John Rosecrans

U.G.L.I.—User Groups of
Lower Idaho
Rt 4
Rupert, ID 83350
Sean Brixey, President

Commodore Users Group
310 Emerald Dr.
Kellogg, ID 83837
(208) 784-8751
Grant Bewick

64-B.U.G. (Boise Users Group)
403 Thatcher St.
Boise, ID 83702
(208) 384-1423
Rick Ohnsman

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee
Avenue
Chicago, IL 60647

VIC 20/64 Users
Support Group
c/o David R. Tarvin
114 S. Clark Street
Pana, IL 62557
(217) 562-4568

Central Illinois PET
User Group
635 Maple
Mt. Zion, IL 62549
(217) 864-5320
Contact: Jim Oldfield

ASM/TED User Group
200 S. Century
Rantoul, IL 61866
(217) 893-4577
Contact: Brant Anderson

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt,
President

Rockford Area PET
Users Group
1608 Benton Street
Rockford, IL 61107

Commodore Users Club
1707 East Main St.
Olney, IL 62450
Contact: David E. Lawless

VIC Chicago Club
3822 N. Bell Ave.
Chicago, IL 60618
John L. Rosengarten

Chicago Commodore 64
Users & Exchange Group
P.O. Box 14233
Chicago, IL 60614
Jim Robinson

Fox Valley PET
Users Group
833 Willow St.
Lake in the Hills, IL 60102
(312) 658-7321
Art DeKneef

The Commodore 64
Users Group Inc.
P.O. Box 46464
Lincolnwood, IL 60646
(312) 583-4629
David Tam Kin

RAP 64/VIC Regional
Assoc. of Programmers
10721 S. Lamon
Oak Lawn, IL 60453
Bob Hughes

The Kankakee Hackers
RR #1, Box 279
St. Anne, IL 60964
(815) 933-4407
Rich Westerman

WIPUG
Rt. 5, Box 75
Quincy, IL 62301
(217) 656-3671
Edward Mills

Papug-Peoria Area Pet
Users Group
6 Apple Tree Lane
East Peoria, IL 61611

(309) 673-6635
Max Taylor
2nd Fri. of Month

McHenry County
Commodore Club
4900 S. Route 31
Crystal Lake, IL 60014
(815) 455-3942
John Katkus
2nd Sat. of month 9 to 12 a.m.

Mt. Vernon Commodore Users
Group (MVCUG)
P.O. Box 512
Mt. Vernon, IL 62864

Commodore 64 Users Club
104 Susan Lane
Carterville, IL 62918
(618) 985-4710
Doyne Horsley

Illinois Valley Commodore
Users Group
2330—12th St.
Peru, IL 61354
(815) 223-5141
Brian Foster

Champaign-Urbana
Commodore Users Group
2006 Crescent Dr.
Champaign, IL 61821
(217) 352-9681
Steve Gast

COMCOE (Commodore Club
of Evanston)
2108 Sherman Ave.
Evanston, IL 60201
Jim Salsbury

Fox Valley 64 Users Group
P.O. Box 28
N. Aurora, IL 60542
(312) 898-2779
Frank Christensen
1st Thursday of month

INDIANA

PET/64 Users
10136 E. 96th St.
Indianapolis, IN 46256
(317) 842-6353
Jerry Brinson

Cardinal Sales
6225 Coffman Road
Indianapolis, IN 46268
(317) 298-9650
Contact: Carol Wheeler

CHUG (Commodore
Hardware Users Group)
12104 Meadow Lane
Oaklandon, IN 46236
Contact: Ted Powell

VIC Indy Club
P.O. Box 11543
Indianapolis, IN 46201
(317) 357-6906
Fred Imhausen

Northern Indiana
Commodore Enthusiasts
927 S. 26th St.
South Bend, IN 46615
Eric R. Bean

Commodore Users Group
1020 Michigan Ave.
Logansport, IN 46947
(219) 722-5205
Mark Bender

Computer Workshop VIC
20/64 Club
282 S. 600 W.
Hebron, IN 46341
(219) 988-4535
Mary O'Bringer

The National Science Clubs
of America
Commodore Users Division
P.O. Box 10621
Merrillville, IN 46411
Brian Lepley or Jeff Brown

East Central Indiana VIC User
Group
Rural Route #2
Portland, IN 47371
Stephen Erwin

National VIC 20 Program
Exchange
102 Hickory Court
Portland, IN 47371
(219) 726-4202
Stephen Erwin

Commodore Computer Club
3814 Terra Trace
Evansville, IN 47711
(812) 477-0739
John Patrick, President

Commodore 64 Users Group
912 South Brown Ave.
Terre Haute, IN 47803
(812) 234-5099
Dennis Graham

Seymour Peekers
c/o D&L Camera Shop
108 N. Chestnut
Seymour, IN 47274
Dennis Peters

VIC/64 Users Group
c/o Delco Remy Div. General
Motors
2401 Columbus Ave.
Anderson, IN 46014
(317) 378-3016
Richard Clifton
3rd Wed. or Thurs. of month

Commodore Owners Of
Lafayette (COOL!)
20 Patrick Lane
West Lafayette, IN 47906
(317) 743-3410
Ross Indelicato

IOWA

Commodore User Group
114 8th St.
Ames, IA 50010

Quad City Commodore Club
P.O. Box 3994
Davenport, IA 52808
(319) 242-1496
Mike Hoeper
3rd Tues.

Siouxland Commodore Club
2700 Sheridan St.
Sioux City, IA 51104
(712) 258-7903
Gary Johnson
1st & 3rd Monday of month

421 W. 6th St.
Waterloo, IA 50702
(319) 232-1062
Frederick Volker

Commodore Computer Users
Group of Iowa
Box 3140
Des Moines, IA 50316
(515) 263-0963 or
(515) 287-1378
Laura Miller

Commo-Hawk Commodore
Users Group
P.O. Box 2724
Cedar Rapids, IA 52406
Vern Rotert

Newton Commodore
Users Group
320 W. 9th St., S.
Newton, IA 50208
(515) 792-0814
David Schmidt
1st Wed.

KANSAS

Wichita Area PET
Users Group
2231 Bullinger
Wichita, KS 67204
(316) 838-0518
Contact: Mel Zandler

Kansas Commodore
Computer Club
101 S. Burch
Olathe, KS 66061
Contact: Paul B. Howard

Commodore Users Group
6050 S. 183 St. West
Viola, KS 67149
Walter Lounsbery

Walnut Valley Commodore
User Group
1003 S. 2nd St.
Arkansas City, KS 67005
Bob Morris

KENTUCKY

VIC Connection
1010 S. Elm
Henderson, KY 42420
Jim Kemp

Louisville Users of
Commodore KY.
(LUCKY)
P.O. Box 22244
Louisville, KY 40222
(502) 425-2847
2nd Tues. of Month

The Bowling Green
Commodore Users Group
Route 11, Creekside Apt. #6
Bowling Green, KY 42101
(502) 781-9098
Alex Fitzpatrick

LOUISIANA

Franklin Parish Computer
Club
#3 Fair Ave.
Winnisboro, LA 71295
James D. Mays, Sr.

NOVA
917 Gordon St.
New Orleans, LA 70117
(504) 948-7643
Kenneth McGruder, Sr.

VIC 20 Users Group
5064 Bowdon St.
Marrero, LA 70072
(504) 341-5305
Wayne D. Lowery, R.N.

64-Club News
5200 Corporate Blvd.
Baton Rouge, LA 70808
(504) 925-5870
Tom Parsons
3rd Tues. of month at CWA

Commodore Users Group
of Oachita
P.O. Box 175
Swaric, LA 71281
(318) 343-8044
Beckie Walker

Ark-La-Tex Commodore 64
Club
198 India Dr.
Shreveport, LA 71115
(318) 797-9702
Pete Whaley

Commodore 64 Users Group
P.O. Box 1422
Baton Rouge, LA 70821
Richard Hood
3rd Tues. of month

MAINE

COM-VICS (Commodore/VIC
Users Group)
RFD #1, Box 2086
Hebron, ME 04238
(207) 966-3641
Paul Lodge
1st Wed. & 3rd Thurs.

Your Commodore Users
Group
Box 611
Westbrook, ME 04092
(207) 854-4579
Mike Procise

Northwoods Commodore
Users Group
740 Main St.
Van Buren, ME 04785
Diane Porter

So. ME. 64
10 Walker St.
Portland, ME 04102
(207) 761-1626
Ed Moore

MARYLAND

Assoc. of Personal
Computer Users
5014 Rodman Road
Bethesda, MD 20016

Blue TUSK
700 East Joppa Road
Baltimore, MD 21204
Contact: Jim Hauff

House of Commodore
8835 Satyr Hill Road
Baltimore, MD 21234
Contact: Ernest J. Fischer

Long Lines Computer Club
323 N. Charles St., Rm. 201
Baltimore, MD 21201
Gene Moff

VIC & 64 Users Group
The Boyds Connection
21000 Clarksburg Rd.
Boyds, MD 20841
(301) 428-3174
Tom DeReggi

Rockville VIC/64 Users Group
P.O. Box 8805
Rockville, MD 20856
(301) 231-7823
Tom Pounds

The Compucats' Commodore
Computer Club
680 W. Bel Air Ave.
Aberdeen, MD 21001
(301) 272-0472
Betty Jane Schueler

Westinghouse BWI
Commodore User Group
Attn: L. Barron Mail Stop
5320
P.O. Box 1693
Baltimore, MD 21203

HUG (Hagerstown Users
Group)
23 Coventry Lane
Hagerstown, MD 21740
(301) 797-9728
Joseph Rutkowski

The Montgomery Ct.
Commodore Computer Soc.
P.O. Box 6444
Silver Springs, MD 20906
(301) 946-1564
Meryle Pounds

Commodore Users Group
of Annapolis
P.O. Box 9726
Arnold, MD 21012
(301) 974-4548
The Software Co.

Gaithersburg C-64 Users Group
12937 Pickering Dr.
Germantown, MD 20874
(301) 428-3328
Russel Jarosinski
3rd Thurs. G'burg Library

Jumpers 2064s
7837 B&A Blvd.
Glen Burnie, MD 21061
(301) 768-1892
Walt Marhefka
Jumpers Mall, 1st Monday

Commodore 64 Users Group
11209 Tack House Court
Potomac, MD 20854
(301) 983-8199
Jorge Montalvan

VIClique (Linthicum Heights)
105A Conduit St.
Annapolis, MD 21401
(301) 263-8568
Pat Foley
M.I.T.A.G.S., 7pm Mon. varys

Southern MD Commodore
Users Group
6800 Killarney St.
Clinton, MD 20735
(301) 868-6536
Tom Helmke
1st Tues. of month 7:30pm

Edison Commodore
Users Group
4314 Oxford Dr.
Suitland, MD 20746
(301) 423-7155
Bill Foley
Naval Research Laboratory

MASSACHUSETTS

Eastern Massachusetts
VIC Users Group
c/o Frank Ordway
7 Flagg Road
Marlboro, MA 02173

VIC Users Group
c/o Ilene Hoffman-Sholar
193 Garden St.
Needham, MA 02192

Commodore Users Club
Stoughton High School
Stoughton, MA 02072
Contact: Mike Lennon

Berkshire PET Lovers
CBM Users Group
Taconic High
Pittsfield, MA 01201

The Boston Computer Society
Three Center Plaza
Boston, MA 02108
(617) 367-8080
Mary E. McCann

Masspet Commodore
Users Group
P.O. Box 307
East Taunton, MA 02718
David Rogers

Raytheon Commodore
Users Group
Raytheon Company
Hartwell Rd. GRA-6
Bedford, MA 01730
John Rudy

Commodore 64 Users
Group of The Berkshires
184 Highland Ave.
Pittsfield, MA 01201
Ed Rucinski

VIC Interface Club
48 Van Cliff Ave.
Brockton, MA 02401
Bernie Robichaud

Cape Cod 64 Users Group
358 Forrest Rd.
S. Yarmouth, MA 02664
1 (800) 225-7136
Jim Close
(In MA. call) 1 (800) 352-7787

The Cursor Club
442 Mulpuf Rd.
Lunenburg, MA 01462
(617) 582-0529
John

Pioneer Valley VIC/64 Club
34 Bates St.
Westfield, MA 01085
(413) 562-1027
Marvin Yale
3rd Thurs. of month

EM 20/64 Users Group
36 Buckman St.
Woburn, MA 01801
John Chaplain

Commodore Users Group
c/o Best Business Equipment
269 Lincoln St.
Worcester, MA 01605

Berkshire Home for Little
PET Users
401 Pomeroy Ave.
Pittsfield, MA 01201
Tim Auxier

Pioneer Valley VIC Club
34 Bates Ave.
Westfield, MA 01085
(413) 562-1027
Marvin Yale

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

VIC Users Club
University of Michigan
School of Public Health
Ann Arbor, MI 48109
Contact: John Gannon

Commodore User Club
32303 Columbus Drive
Warren, MI 48093
Contact: Robert Steinbrecher

Commodore Users Group
c/o Family Computer
3947 W. 12 Mile Rd.
Berkley, MI 48072

VIC for Business
6027 Orchard Ct.
Lansing, MI 48910
Mike Marotta

South Computer Club
South Jr. High School
45201 Owen
Belleville, MI 48111
Ronald Ruppert

Commodore Users Group
c/o Eaton Rapids Medical Clinic
101 Spicerville Hwy.
Eaton Rapids, MI 48827
Albert Meinke III, M.D.

South East Michigan Pet
Users Group
Box 214
Farmington, MI 48024
Norm Eisenberg

Commodore Computer Club
4106 Eastman Rd.
Midland, MI 48640
(517) 835-5130
John Walley
9:30 p.m. Sept/May

VIC, 64, PET Users Group
8439 Arlis Rd.
Union Lake, MI 48085
363-8539
Bert Searing

COMP
486 Michigan Ave.
Marysville, MI 48040
(313) 364-6804
M. Gauthier

Ann Arbor Commodore
Users Group
Ann Arbor, MI 48103
(313) 994-4751
Art Shaw
3rd Tues. 7:30-10:00

DAB Computer Club
P.O. Box 542
Watervliet, MI 49098
(616) 463-5457
Dennis Burlingham

West Michigan Commodores
c/o R. Taber
1952 Cleveland Ave., S.W.
Wyoming, MI 49509
(616) 458-9724
Gene Traas

Debug
P.O. Box 196
Berrien Springs, MI 49103
(616) 471-1882
Herbert Edward
Last Thursday of Month

Jackson Commodore
Computer Club
201 S. Grinnell St.
Jackson, MI 49203
Alfred Bruey
Last Thur. of Month 7:30 p.m.

SMCUG
1002 Pfau St.
Mankato, MI 56001
(507) 625-6942
Dean Otto

SEM 64
25015 Five Mile #3
Redford, MI 48239
(313) 537-4163
Gary Groeller

C.A.T.O.
17606 Valade
Riverview, MI 48192
Dean Tidwell

Mid-Michigan
Commodore Club
Clare, MI
(517) 386-3429
Virgil Graham
3rd Mon. 7pm Clare H.S.

Michigan's Commodore 64
Users Group (MCUG)
P.O. Box 539
E. Detroit, MI 48021
(313) 773-6302
William G. Osipoff

Michigan's Commodore 64
Users Group
P.O. Box 539
East Detroit, MI 48021

20050 Winchechester
Southfield, MI 48076
(313) 354-7224
Steve Lepsetz 353-1130

MINNESOTA

MUPET (Minnesota Users
of PET)
P.O. Box 179
Annandale, MN 55302
c/o Jon T. Minerich

Twin Cities Commodore
Computer Club
6623 Ives Lane
Maple Grove, MN 55369
(612) 424-2425
Contact: Rollie Schmidt

Brainerd Area Commodore
Users Group
1219 S.E. 11th St.
Brainerd, MN 56401

(218) 829-0805
Norm Saavedra
1st Thurs. 6 p.m. & 3rd Sat. 10 a.m.

Lake Superior Commodore
1936 Lawn St.
Duluth, MN 55812
(218) 728-3224
Peter Roufs

Heartland Area Computer Cooperative
Route 4, Box 204
Little Falls, MN 56345
(612) 632-5511
Robert Walz

MISSISSIPPI

Commodore Biloxi User Group (ComBUG)
Universal Computer Services
3002 Hwy. 90 East
Ocean Springs, MS 39564
(601) 875-1173
John Lassen

Commodore Computer Club
Southern Station Box 10076
Hattiesburg, MS 38401
(601) 268-7585
Andrew Holder

Commodore Biloxi Users Group
c/o Universal Computer Services
3002 Hwy. 90 East
Ocean Springs, MS 39564
(601) 875-1173
John Lassen

MISSOURI

KCPUG
(Commodore User Group of Kansas City)
P.O. Box 36492
Kansas City, MO 64111
(816) 252-7628
Salvadore

Commodore User Group of St. Louis
Box 6653
St. Louis, MO 63125-0653
Dan Weidman, New Members
1541 Swallowtail Dr.
St. Louis, MO

VIC INFONET
P.O. Box 1069
Branson, MO 65616
(417) 334-6099
Jory Sherman

Worth County PET Users Group
Grant City, MO
(816) 564-3551
David Hardy

Mid-Missouri Commodore Club
780 E. Park Lane
Columbia, MO 65201
(314) 474-2868
Jim Whitacre

Joplin Commodore Computers Users Group
422 S. Florida Ave.
Joplin, MO 64801
R. D. Connely

MOARK Commodore Users Group
P.O. Box 504
Golden, MO 65658
(417) 271-3293
Marshall Turner

MONTANA

Powder River Computer Club
Powder River County High School
Broadus, MT 59317
Contact: Jim Sampson

Commodore User Club
1109 West Broadway
Butte, MT 59701
Contact: Mike McCarthy

NEBRASKA

Greater Omaha Commodore 64 Users Group
2932 Leawood Dr.
Omaha, NE 68123
(402) 292-2753
Bob Quisenberry

NEVADA

Las Vegas PET Users
Suite 5-315
5130 E. Charleston Blvd.
Las Vegas, NV 89122
Gerald Hasty

Compu Club 64
4220 S. Maryland Parkway
Bldg. B—Suite 403
Las Vegas, NV 89109
(702) 369-7354
Cindy Springfield

NEW JERSEY

Commodore Friendly User Group
49 Hershey Rd.
Wayne, NJ 07470
(201) 696-8043
Rich Pinto/Colin Campbell

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

Educators Advisory
P.O. Box 186
Medford, NJ 08055
(609) 953-1200
John Handfield

VIC-TIMES
46 Wayne Street
Edison, NJ 08817
Thomas R. Molnar

VIC 20 User Group
67 Distler Ave.
W. Caldwell, NJ 07006
(201) 284-2281
G. M. Amin

VIC Software Development Club
77 Fomalhaut Ave.
Sewell, NJ 08080
H. P. Rosenberg

ACGNJ PET/VIC/CBM User Group
30 Riverview Terr.
Belle Mead, NJ 08502
(201) 359-3862
J. M. Pylka

South Jersey Commodore Users Club
46-B Monroe Path
Maple Shade, NJ 08052
(609) 667-9758
Mark Orthner

Parsippany Computer Group
51 Ferncliff Rd.
Morris Plains, NJ 07950
(201) 267-5231
Bob Searing

Information 64
16 W. Ridgewood Ave.
Ridgewood, NJ 07450
(201) 447-4432
Dave Garaffa

NEW HAMPSHIRE

Northern New England Computer Society
P.O. Box 69
Berlin, NH 03570

TBH VIC-NICs
P.O. Box 981
Salem, NH 03079

C-64 U.S.E.R.S. User Software Exchange Pro
P.O. Box 4022
Rochester, NH 03867
Paul Kyle

NEW MEXICO

Commodore Users Group
6212 Karlson, NE
Albuquerque, NM 87113
(505) 821-5812
Danny Byrne

NEW YORK

Capital District 64/VIC 20 Users Group
363 Hamilton St.
Albany, NY 12210
(518) 436-1190
Bill Pizer

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club of Westchester
P.O. Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

Commodore Masters
25 Croton Ave.
Staten Island, NY 10301
Contact: Stephen Farkouh

VIC Users Club
76 Radford St.
Staten Island, NY 10314
Contact: Michael Frantz

West Chester County VIC Users Group
P.O. Box 146
Pelham, NY 10552
Joe Brown

SPUG
4782 Boston Post Rd.
Pelham, NY 10803
Paul Skipski

VIC 20 User Club
151-28 22nd Ave.
Whitestone, NY 11357
Jean F. Coppola

VIC 20 User Club
339 Park Ave.
Babylon, NY 11702
(516) 669-9126
Gary Overman

VIC User Group
1250 Ocean Ave.
Brooklyn, NY 11230
(212) 859-3030
Dr. Levitt

L&M Computer Club
VIC 20 & 64
4 Clinton St.
Tully, NY 13159
(315) 696-8904
Dick Mickelson

Commodore Users Group
1 Corwin Pl.
Lake Katrine, NY 12449
J. Richard Wright

VIC 20/Commodore 64 Users Group
31 Maple Dr.
Lindenhurst, NY 11757
(516) 957-1512
Pete Lobol

VIC Information Exchange Club
336 W. 23 St.
Deer Park, NY 11729
Tom Schlegel
SASE & phone please

New York Commodore Users Group
380 Riverside Dr., 7Q
New York, NY 10025
(212) 566-6250
Ben Tunkelang

Hudson Valley Commodore Club
1 Manor Dr.
Woodstock, NY 12498
F.S. Goh
1st Wednesday of month

LIVICS (Long Island VIC Society)
20 Spyglass Lane
East Setauket, NY 11733
(516) 751-7844
Lawrence Stefani

VIC Users Group
c/o Stoney Brook Learning Center
1424 Stoney Brook Rd.
Stoney Brook, NY 11790
(516) 751-1719
Robert Wurtzel

Poughkeepsie VIC User Group
2 Brooklands Farm Rd.
Poughkeepsie, NY 12601
(914) 462-4518
Joe Steinman

VIC 20 User Group
Paper Service Division
Kodak Park
Rochester, NY 14617
David Upham, Sr.

Manhattan 64
c/o Steve Lazarowitz
1440 Freeport Loop
Brooklyn, NY 11239
(212) 647-4266
Larry Thompson

Adirondack Commodore 64 Users Group
205 Woodlawn Ave.
Saratoga Springs, NY
(518) 584-8960
Paul Klompas

Rockland County Commodore Users Group
P.O. Box 573
Nanuet, NY 10965
Ross Garber

New York 64 Users Group
222 Thompson St.
New York, NY 10012
(212) 673-7241
Bruce Cohen

Finger Lakes Commodore Users Group
c/o Rose City Computer Associates
229 West Union St.
Newark, NY 14513
(315) 331-1185

The Commodore Users Group Rochester
78 Hardison Rd.
Rochester, NY 14617
(716) 544-5251
Tom Werenski
Phone Evenings between 7-10

Commodore Computer Club
Publications Dept.,
Grumman Aerospace
1111 Stewart Ave.
Bethpage, NY 11714
(516) 575-9558
Neil Threulsen

Hello, Central!
76-12 35th Ave.
Jackson Heights, NY 11372
Jared Sherman

VIC 20/64 Users Group
NYU Waverly Place
New York, NY 10003
(212) 358-5155
Lawrence Schulman

SCUG (Schenectady Commodore Users Group)
c/o the Video Connection
Canal Square
Schenectady, NY 12305
Timothy Davis
1st Mon. of Month

Commodore 64 Users Group
S.U.N.Y. at Oswego
Dept. of Industrial Arts
Oswego, NY 13126
John R. Boronkay

NORTH CAROLINA

Amateur Radio PET Users Group
P.O. Box 30694
Raleigh, NC 27622
Contact: Hank Roth

VIC Users Club
c/o David C. Fonenberry
Route 3, Box 351
Lincolnton, NC 28092

Microcomputer Users Club
Box 17142 Bethabara Sta.
Winston-Salem, NC 27116
Joel D. Brown

VIC Users Club
Rt. 11, Box 686
Hickory, NC 28601
Tim Gromlovits

Raleigh VIC 20/64 Users Group
410-D Delta Court
Cary, NC 27511
(919) 469-3862
Larry Diener

Down East Commodores
302 Belltown Rd.
Havelock, NC 28532
(919) 447-4536
Bruce Thedin

Tryon Commodore 64 Club
P.O. Box 1016
Tryon, NC 28782
(704) 859-6340
Robin Michael
1st Mon. of month at 7 p.m.

OHIO

Dayton Area PET
User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, President
(513) 848-2065
J. Watson, Secretary
(513) 372-2052

Central Ohio PET
Users Group
107 S. Westmoor Avenue
Columbus, OH 43204
(614) 274-0304
Contact: Philip H. Lynch

Commodore Computer Club
of Toledo
734 Donna Drive
Temperance, MI 48182
Gerald Carter

Chillicothe Commodore
Users Group
P.O. Box 211
Chillicothe, OH 45601
William A. Chaney

Licking County 64 Users Group
323 Schuler St.
Newark, OH 43055
(614) 345-1327

11433 Pearl Rd.
Strongsville, OH 44136
Paul M. Warner

C.P.U. Connection
P.O. Box 42032
Brook Park, OH 44142
Danni Hudak

SE Cleveland Commodore
Crazies
18813 Harlan Dr.
Maple Heights, OH 44137
(216) 581-3099
Carl Skala

Commodore Users of
Blue Chip (Cincinnati)
816 Beecher St.
Cincinnati, OH 45206
(513) 961-6582
Ted Stalets

Southwestern Ohio
Commodore Users Group
P.O. Box 399117
Cincinnati, OH 45239
2nd Wed. of month at 7 p.m.

S.W.O.C.U.G. (SW. Ohio
Commodore Users Gp.)
8401 Wicklow Ave.
Cincinnati, OH 45236
Joe Beresford

OKLAHOMA

Southwest Oklahoma
Computer Club
c/o Commodore Chapter
P.O. Box 6646
Lawton, OK 73504
1:30 at Lawton City Library

Tulsa Area Commodore
Users Group
Tulsa Computer Society
P.O. Box 15238
Tulsa, OK 74112
Annette Hinshaw

Commodore Oklahoma
Users Club
4000 NW 14th St.
Oklahoma City, OK 73107
(405) 943-1370
Stanley B. Dow

Commodore Users
Box 268
Oklahoma City, OK 73101
Monte Maker, President

Commodore Users of Norman
209 Brookwood
Noble, OK 73068
Matt Hager

Commodore Users Group
Muskogee Computer Society
202 S. 12th St.
Muskogee, OK 74401
Steve Ford

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

U.S. Commodore Users Group
P.O. Box 2310
Roseburg, OR 97470
(503) 672-7591
Richard Tsukiji

Southern Oregon VIC/64
Users Group
3600 Madrona Lane
Medford, OR 97501
(503) 779-7631
James Powell

PENNSYLVANIA

PET User Group
Gene Beals
P.O. Box 371
Montgomeryville, PA 18936

Penn Conference
Computer Club
c/o Penn Conference of SDA
720 Museum Road
Reading, PA 19611
Contact: Dan R. Knepp

PACS Commodore Users Group
LaSalle College
20th & Olney Ave.
Philadelphia, PA 19141
(215) 951-1258
Stephen Longo

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

PPG (Pittsburgh PET Group)
c/o Joel A. Casar, DMD
2015 Garrick Drive
Pittsburgh, PA 15235
(412) 371-2882

Commodore Users Club
3021 Ben Venue Dr.
Greensburg, PA 15601
(412) 836-2224
Jim Mathers

VIC 20 Programmers, Inc.
c/o Watson Woods
115 Old Spring Rd.
Coatesville, PA 19320
Robert Gougher

G.R.C. User Club
300 Whitten Hollow Rd.
New Kensington, PA 15068
Bill Bolt

NADC Commodore Users Club
248 Oakdale Ave.
Horsham, PA 19044
Norman McCrary

CACC (Capitol Area
Commodore Club)
134 College Hill Rd.
Enola, PA 17025
(717) 732-2123
Lewis Buttery
Union Deposit Mall at 7 p.m.

G/C Computer Owners Group
c/o Gilbert Associates, Inc.
P.O. Box 1498
Reading, PA 19607
Extension 6472
Jo Lambert (215) 775-2600

Boeing Employees Personal
Computer Club
The Boeing Vertol Co.
P.O. Box 16858
Philadelphia, PA 19142
(215) 522-2257
Jim McLaughlin

South Central PA
Commodore Club
2109 Cedar Run Dr.
Camp Hill, PA 17011
(717) 763-4219
David Persing

Main Line Commodore Users
Group (MLCUG)
c/o Main Line Computer Center
1046 General Allen Lane
West Chester, PA 19380
(215) 388-1581
Emil Volcheck

Commodore Users Group
781 Dick Ave.
Warminster, PA 18974
Matt Matulaitis

The Commodore Users Club
of S.E. Pittsburgh
c/o Groves Appliance & TV
2407 Pennsylvania Ave.
West Mifflin, PA 15122
Charles Groves

Compstars
130 Blue Teal Circle
Audubon, PA 19403
Mike Norm
Meet at Audio Video Junct.

Scranton Commodore
Users Group
P.O. Box 211
Clarks Summit, PA 18411

Clifton Heights Users Group
P.O. Box 235
Clifton Heights, PA 19018

Oxford Circle 64 Users Group
Frankford Cong. Un. Church
of Christ
Oxford Ave. & Pratt St.
Philadelphia, PA 19124
(215) 743-8999
Roger Nazeley (215) 535-9021
4th Wed. of Month

VIC Software
Development Club
440 W. Sedgwick
Apt. A-1
Philadelphia, PA 19119
(215) 844-4328
Tracy Lee Thomas

Bits & Bytes
1015 Dale Rd.
Secane, PA 19018
(215) 544-5875
Dave Boodey

CACCC-Centre Area
Commodore Computer Club
214 Computer Building
University Park, PA 16802
(814) 237-5912
Bill Hillner

PUERTO RICO

CUG of Puerto Rico
RFD #1, Box 13
San Juan, PR 00914
Ken Burch

VIC 20 User Group
655 Hernandez St.
Miramar, PR 00907
Robert Morales, Jr.

RHODE ISLAND

Irving B. Silverman, CPA
160 Taunton Ave.
E. Providence, RI 02914
Contact: Michelle Chavanne

Newport VIC/64 Users
10 Maitland Ct.
Newport, RI 02840
(401) 849-2684
Dr. Matt McConeghy

The VIC 20 Users Club
Warwick, RI 02886
Tom Davey

Commodore Users Group
c/o Data-Co.
978 Tiogue Ave.
Coventry, RI 02816
(401) 828-7385
Victor Moffett

SOUTH CAROLINA

Beaufort Technical College
100 S. Ribaut Rd.
Beaufort, SC 29902
Dean of Instruction

Computer Users Society
of Greenville (CUS)
Horizon Records-
Home Computers
347 S. Pleasantburg Dr.
Greenville, SC 29607
(803) 235-7922
Bo Jeanes

Commodore Computer
Club of Columbia
318 Quincannon Dr.
Columbia, SC 29210
Buster White Sect./Treas.

Spartanburg Commodore
Users Group
803 Lucerne Dr.
Spartanburg, SC 29302
(803) 582-5897
James Pasley

SOUTH DAKOTA

PET User Group
515 South Duff
Mitchell, SD 57301
(605) 996-8277
Contact: Jim Dallas

VIC/64 Users Club
608 West 5th
Pierre, SD 57501
(605) 224-4863
Larry Lundeen

TENNESSEE

River City Computer
Hobbyists
Memphis, TN
1st Mon. at Main Library

Nashville Commodore
Users Group
P.O. Box 121282
Nashville, TN 37212
(615) 331-5408
Dave Rushing
3rd Thurs. at Cumberland Mus

Commodore User Club
Metro Computer Center
1800 Dayton Blvd.
Chattanooga, TN 37405
Mondays 7:30 pm

Metro-Knoxville Commodore
Users Club
7405 Oxmoor Rd., Rt. #20
Knoxville, TN 37921
(615) 938-3773
Ed Pritchard

Memphis Commodore
Users Group
2476 Redvers Ave.
Memphis, TN 38127
(901) 358-5823
Harry Ewart

TEXAS

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
P.OD Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M
Microcomputer Club
Texas A & M, TX

CHUG (Commodore Houston
Users Group)
8738 Wildforest
Houston, TX 77088
(713) 999-3650
Contact: John Walker

Commodore Users Group
5326 Cameron Rd.
Austin, TX 78723
(512) 459-1220
Dr. Jerry D. Frazee

VIC Users Group
3817 64th Dr.
Lubbock, TX 79413

64 Users Group
2421 Midnight Circle
Plano, TX 75075
S.G. Grodin

Savid Computer Club
312 West Alabama
Suite 2
Houston, TX 77006
Davi Jordan, Chairman

Gulf Coast Commodore
Users Group
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577
Lawrence Hernandez

Mid-Cities Commodore Club
413 Chisolm Trail
Hurst, TX 76053
Garry Wordelman

Mid-Cities Commodore Club
413 Chisolm Trail
Hurst, TX 76053
Bruce Nelson

Interface Computer Club
814 North Sabinas
San Antonio, TX 78207
M. E. Garza, President

Gulf Coast Commodore
Users Group
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577
Lawrence Hernandez

ICUG (Irving Commodore
Users Group)
3237 Northgate #1289
Irving, TX 75062
(214) 252-7017
Robert Hayes

Commodore Computer
Club (C3)
c/o Lamar Full Gospel
Assembly
1200 S. Sumner
Pampa, TX 79065
(806) 665-3444
Randy Mills
Every other Thurs. 7 p.m.

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore Users
Club
742 Taylor Avenue
Ogden, UT 84404
Contact: Todd Woods Kap,
President
David J. Shreeve,
Vice President

The VIClic
799 Ponderosa Drive
Sandy, UT 84070
Contact: Steve Graham

VIC 20 Users
324 N. 300 W.
Smithfield, UT 84335
Dave DeCorso

Northern Utah VIC & 64
Users Group
P.O. Box 533
Garland, UT 84312
David Sanders

The Commodore Users Group
652 West 700 North
Clearfield, UT 84015
(801) 776-3950
Rodney Keller,
Richard Brenchly

Mountain Computer Society
P.O. Box 1154
Sandy, UT 84091
Dave Tigner

VIRGINIA

Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(803) 860-9116

VIC Users Group
Rt. 2, Box 180
Lynchburg, VA 24501
Contact: Dick Rossignol

VIC Users Group
c/o Donnie L. Thompson
1502 Harvard Rd.
Richmond, VA 23226

Dale City Commodore
User Group
P.O. Box 2004
Dale City, VA 22193
(703) 680-2270
James Hogler

Tidewater Commodore
Users Group
4917 Westgrove Rd.
Virginia Beach, VA 23455
Fred Monson

Fredericksburg Area
Computer Enthusiasts
P.O. Box 324
Locust Grove, VA 22508
(703) 972-7195
Michael Parker

VIC 20 Victims
4301 Columbia Pike #410
Arlington, VA 22204
(703) 920-0513
Mike Spengel

Peninsula Commodore 64
Users Group
124 Burnham Place
Newport News, VA 23606
(804) 595-7315
Richard G. Wilmoth

Norfolk Users Group
1030 West 43rd St. B-4
Norfolk, VA 23508
489-8292
Larry Pearson

NASA VIC 20 User Group
713 York Warwick Dr.
Yorktown, VA 23692
Harris Hamilton

135 Beverley Rd.
Danville, VA 24541
David Gray

R.A.C.E. Commodore
Users Group
4726 Horseman Dr.
Roanoke, VA 24019
(703) 362-3960
Larry Rackow

Commodore Users of Franklin
1201 N. High St.
Franklin, VA 23851
(804) 562-6823
D. Bruce Powell

WASHINGTON

NW PET Users Group
2565 Dexter N. 3203
Seattle, WA 98109
Contact: Richard Ball

PET Users Group
c/o Kenneth Tong
1800 Taylor Ave. N102
Seattle, WA 98102

Whidbey Island Commodore
Computer Club
947 N. Burroughs Ave.
Oak Harbor, WA 98277
Michael D. Clark

Central Washington
Commodore Users Group
1222 S. 1st St.
Yakima, WA 98902
Tim McElroy

Blue Mountain Commodore
Users Club
15 Stone St.
Walla Walla, WA 99362
(509) 525-5452
Keith Rodue

Spokane Commodore
User Group
N. 4311 Whitehouse
Spokane, WA 99205
(509) 328-1464
Stan White

CBM Users Group
803 Euclid Way
Centralia, WA 98531
(206) 736-4085
Rick Beaber

Computer Club
c/o Honeywell, Inc.
5303 Shilshole Ave., NW
Seattle, WA 98107
(206) 789-2000
John Goddard

WEST VIRGINIA

Personal Computer Club
P.O. Box 1301
Charleston, WV 25325
Cam Cravens

TriState Commodore Users
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

Commodore User Group
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

WISCONSIN

Sewpus
c/o Theodore J. Polozynski
P.O. Box 21851
Milwaukee, WI 53221

Waukesha Area Commodore
User Group (WACUG)
256½ W. Broadway
Waukesha, WI 53186
Contact: Walter Sadler
(414) 547-9391

Commodore 64 Software
Exchange Group
P.O. Box 224
Oregon, WI 53575
E. J. Rosenberg

C.L.U.B. 84
6156 Douglas Ave.
Caledonia, WI 53108
(414) 835-4645 pm
Jack White
2nd Sat every month 10:00 am

VIC-20 & 64 User Group
522 West Bergen Dr.
Milwaukee, WI 53217
(414) 476-8125
Mr. Wachtl

Menomonie Area Commodore
Users Group
510 12th St.
Menomonie, WI 54751
(715) 235-4987
Mike Williams

C.U.S.S.H.
3614 Sovereign Dr.
Racine, WI 53406
(414) 554-0156
Tim Tremmel
3rd Saturday of month

Madison Area Commodore
Users Group
1552 Park St.
Middleton, WI 53562
(608) 831-4852
John Carvin
3rd Thurs. each month

S.W.I.T.C.H.
W156 N8834 Pilgrim Rd.
Menomonee Falls, WI 53051
(414) 255-7044
Len Lutz

Milwaukee Area CBM64
Enthusiasts (M.A.C.E.)
P.O. Box 340
Elm Grove, WI 53122
(414) 259-5991
Kevin Wilde

The Eau Claire CBM64
Users Group
Rt. 5, Box 179A
Eau Claire, WI 54703
(715) 874-5972
John Slavsky, Jr.
2nd Thurs. 7 p.m.

WAVE
P.O. Box 0641
Waukesha, WI 53187

WYOMING

Commodore Users Club
c/o Video Station
670 North 3rd #B
Laramie, WY 82070
(307) 721-5908
Pamela Nash

AUSTRALIA

WA VIC-Ups (VIC 20/CBM
64 Users)
14 Glengariff Dr.
Floreat Park 6014
Western Australia
B. J. Cook

AUSTRIA

Commodore Users Club
Postfach 5026
Salzburg
Austria
062-222-5391
D.A. Stagg

CANADA

Toronto PET
Users Group, Inc.
1912A Avenue Rd., Ste. 1
Toronto, Ontario, Canada
M5M 4A1
(416) 782-8900
or call 416-782-9252
Contact: Chris Bennett

PET Users Club
c/o Mr. Brown
Valley Heights
Secondary School
Box 159
Langton, Ont. N0E 1G0

Vancouver PET Users Group
P.O. Box 91164
West Vancouver, British
Columbia
Canada V7V 3N6

CCCC (Canadian
Commodore Computer Club)
c/o Strictly Commodore
47 Coachwood Place
Calgary, Alberta, Canada
T3H 1E1
Contact: Roger Olanson

W.P.U.G.
9-300 Enniskillen Ave.
Winnipeg, Manitoba R2V 0H9
Larry Neufeld

VIC-TIMS
2-830 Helena St.
Trail, British Columbia
V1R 3X2
(604) 368-9970
Greg Goss

Arva Hackers
Medway High School
Arva, Ontario N0M 1C0
D. Lerch

Nova Scotia Commodore
Computer Users Group
66 Landrace Cres.
Dartmouth, N.S. B2W 2P9
Andrew Cornwall

Bonnyville VIC Cursors
Box 2100
Bonnyville, Alberta T0A 0L0
(403) 826-3992
Ed Wittchen

Commodore Users Club
of Sudbury
938 Brookfield Ave.
Sudbury, Ontario
P3A 4K4

PET Educators Group
P.O. Box 454
Station A
Windsor, Ontario
N9A 6L7

COMVIC
P.O. Box 1688
St. Laurent
Montreal, Quebec
H4L 4Z2

Calgary Commodore
Users Group
37 Castleridge Dr., N.E.
Calgary, Alberta
T3J 1P4
John Hazard

Fledging Barrie User
Group (BUG)
58 Steel St.
Barrie, Ontario
Canada L4M 2E9

FINLAND

VIC-Club in Helsinki
c/o Matti Aarnio
Linnustajankj 2B7
SF-02940 ESP00 94
Finland

GERMANY

Kettenberg 24
D 5880 Lueden Scheid
West Germany
Rudi Ferrari

ITALY

Commodore 64 Club
Universita di Studi shan
V. Avigliana 13/1
10138 TORINO
ITALY

KOREA

Commodore Users Club
K.P.O. Box 1437
Seoul, Korea
Contact: S. K. Cha

MEXICO

Asociacion De Usarios Commodore
c/o Alejandro Lopez Arechiga
Holbein 174-6° Piso
Mexico 18, D.F.

Club de Usarios Commodore
Sigma del Norte
Mol del Valle, Local 44
Garza Garcia, N.L. 66220

Club Microvic
Villaldama 225
Col. Chapultepec
Monterrey, N.L.
Mexico 66450
Oscar Sosa, President

NEW ZEALAND

Commodore Users Group
Meet at VHF Clubrooms
Hazel Ave.
Mount Roskill
3rd Wed. of month, 7:30 pm
Roger Altena 278-5262

Nelson VIC Users Group
c/o P.O. Box 860
Nelson, New Zealand
Peter Archer

E.R. Kennedy
c/o New Zealand Synthetic Fuels Corp. Ltd.
Private Bag
New Plymouth

NORWAY

VIC Club of Norway
Nedre Bankegt 10,
1750 Halden
Norway

UNITED KINGDOM

North London Hobby Computer Club
Dept. of Electronics & Communications Engineering
The Polytechnic of North London
Holloway Rd.
London N7 8DB

Croydon Microcomputer Club
111 Selhurst R.
Selhurst, London SE25 6LH
01-653-3207
Vernon Gifford

User Bulletin Board

User group forming in Wheeling, West Virginia. Contact David Hoskins, president, at 304-242-2605 or Alice Shipley, secretary, at 304-242-8362.

Commodore 64 software exchange, for original programs only. Contact David Morell, 404 Brookside Drive, Chapel Hill, NC 27514.

departments
advertisers' index

	Page No.	Reader Service No.
Academy Software	25	9
Batteries Included	9	2
Bible Research Systems	23	8
Cardco	IBC	27
Cheatsheet Products	83	16
Commodore	IFC & 1, 4 & 5, 13, 58 & 59, 62 & 63	
Continental Press	10	3
Covox	23	6
Cow Bay Computing	10	4
Dennison Computer Supplies	BC	28
Generic, Inc.	83	17
Handic Software	61	13
Indus/Tool	105	20
Intelligent Software	105	19
Limbic Systems	29	10
Micro-80	23	7
Microware Distributing	111	22
Microware Distributing	117	24
Midwest Software	117	23
Orange Micro	57	12
PEEK Magazine	110	21
Protecto Enterprizes	72-75	15
Public Domain	128	25
Rockware Data Corporation	86	18
St. Croix Valley Electronics	128	26
Sota Enterprises	33	
SubLOGIC	7	1
SubLOGIC	11	5

You asked for it . . . CARDCO has it!

Another exclusive from Cardco!

The LQ/1 (Letter Quality) PRINTER

In response to your demand and need, CARDCO now has available for the **first** time . . . a "Commodore-ready "LETTER QUALITY PRINTER". Just plug it in and print.

Offering standard friction-feed and optional tractor-feed, the CARDCO LQ/1 PRINTER comes complete with built-in interfacing for all Commodore Personal Computers, as well as compatible input for PC, PC jr., TRS-80 and other personal computers.

"CARDPRINT" C/?+G PRINTER INTERFACE with Graphics

For printers that are not Commodore-ready, Cardco offers the C/?+G PRINTER INTERFACE including all cables and connectors for the following printers: all new Epson MX, RX and FX series, Star Gemini 10X and Delta 10, Prowriter, C.-Itoh 8510, NEC 8023, Okidata 82, 83, 84, 92, 93 and 94, Mannesman Tally Spirit and MT-160, Seikosha GX-100, BMC BX-80 and the Gorilla Banana. Prints the full Commodore character set; graphics, characters, reversed characters and reversed graphics characters.

"WRITE NOW" WORD PROCESSOR SOFTWARE

An excellent time saver, CARDCO OFFERS THE CØ2 "Write Now" program with built-in 80 column display. You see exactly what will print. All special codes can be transmitted to printers maintaining justification. Easy full-screen editing; works with any printer.

"MAIL NOW" MAILING LIST SOFTWARE

CARDCO D/Ø1 quickly (in memory) sorts by zip, category, name and state; fully compatible with "WRITE NOW". Other fine features include: User-oriented; menu-driven operation; each disk supports 600 entries. Format can print single, double or triple labels across.

All Cardco Products are available at your local dealers. Write for illustrated literature.

cardco, inc.
313 Matheson Wichita, Kansas 67214 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

ISBN 0-88731-010-9